

Approximate resilience, monotonicity, and the complexity of agnostic learning

Dana Dachman-Soled
University of Maryland

Vitaly Feldman
IBM Research - Almaden

Li-Yang Tan*
Columbia University

Andrew Wan
Simons Institute, UC Berkeley

Karl Wimmer†
Duquesne University

Abstract

A function f is d -resilient if all its Fourier coefficients of degree at most d are zero, i.e. f is uncorrelated with all low-degree parities. We study the notion of *approximate resilience* of Boolean functions, where we say that f is α -approximately d -resilient if f is α -close to a $[-1, 1]$ -valued d -resilient function in ℓ_1 distance. We show that approximate resilience essentially characterizes the complexity of agnostic learning of a concept class \mathcal{C} over the uniform distribution. Roughly speaking, if all functions in a class \mathcal{C} are far from being d -resilient then \mathcal{C} can be learned agnostically in time $n^{O(d)}$ and conversely, if \mathcal{C} contains a function close to being d -resilient then agnostic learning of \mathcal{C} in the statistical query (SQ) framework of Kearns has complexity of at least $n^{\Omega(d)}$. This characterization is based on the duality between ℓ_1 approximation by degree- d polynomials and approximate d -resilience that we establish. In particular, it implies that ℓ_1 approximation by low-degree polynomials, known to be sufficient for agnostic learning over product distributions, is in fact necessary.

Focusing on monotone Boolean functions, we exhibit the existence of near-optimal α -approximately $\tilde{\Omega}(\alpha\sqrt{n})$ -resilient monotone functions for all $\alpha > 0$. Prior to our work, it was conceivable even that every monotone function is $\Omega(1)$ -far from any 1-resilient function. Furthermore, we construct simple, explicit monotone functions based on Tribes and CycleRun that are close to highly resilient functions. Our constructions are based on general resilience analysis and amplification techniques we introduce. These structural results, together with the characterization, imply nearly optimal lower bounds for agnostic learning of monotone juntas, a natural variant of the well-studied junta learning problem. In particular we show that no SQ algorithm can efficiently agnostically learn monotone k -juntas for any $k = \omega(1)$ and any constant error less than $1/2$.

1 Introduction

The agnostic learning framework [Hau92, KSS94], models learning from examples in the presence of worst-case noise. In this framework the learning algorithm is given random examples $(\mathbf{x}, f(\mathbf{x}))$ where \mathbf{x} is chosen from some distribution D and f is an *arbitrary* Boolean function. The goal of the agnostic learning algorithm for a concept class \mathcal{C} is to output a hypothesis h that agrees with f almost as well as the best function in \mathcal{C} ; that is:

$$\Pr_D[h(\mathbf{x}) \neq f(\mathbf{x})] \leq \min_{c \in \mathcal{C}} \Pr_D[c(\mathbf{x}) \neq f(\mathbf{x})] + \varepsilon,$$

where ε is an error parameter given to the algorithm.

*Supported by NSF grants CCF-1115703 and CCF-1319788.

†Supported in part by NSF-CCF-1117079. Most of this work was done while the author was visiting Simons Institute for the Theory of Computing, University of California-Berkeley

Understanding the complexity of learning in the agnostic model is central to both theory and practice in machine learning research. Learning in this model is notoriously hard, and despite two decades of intensive research our formal understanding of the complexity of agnostic learning is still very limited. Even when D is the uniform distribution over $\{-1, 1\}^n$, agnostic learning has proven extremely challenging: few non-trivial classes are known to be learnable agnostically. The primary technique used for agnostic learning in this setting is the polynomial ℓ_1 regression algorithm introduced in the influential work of Kalai et al [KKMS08]. This algorithm finds a low-degree polynomial that minimizes the ℓ_1 distance to the target function, and can be applied to agnostically learn classes which are well approximated by polynomials. This approach has led to the first agnostic learning algorithm for AC^0 circuits (in quasi-polynomial time) and halfspaces (in $n^{O(1/\varepsilon^2)}$ time) over the uniform distribution [KKMS08] and was used in many other agnostic learning results.

In this work we address the complexity of agnostic learning relative to the uniform and, more generally, product distributions. In addition to running time, a critical but often unstated parameter in lower bounds on agnostic learning is the value of $\text{OPT}_{\mathcal{C}}(D, f) = \min_{c \in \mathcal{C}} \Pr[c(x) \neq f(x)]$ to which the lower bound applies (note that OPT is essentially the noise rate). If a hardness result requires learning functions f for which $\text{OPT}_{\mathcal{C}}(D, f)$ is close to $1/2$, then it does not apply to most practical learning applications. (If \mathcal{C} does not have any useful classifiers, it does not make much sense to use \mathcal{C} as a performance benchmark.) Therefore it is more important to understand the complexity of agnostic learning in which OPT is a small constant close to 0 (or even approaches 0 as n grows). However essentially all known lower bounds for agnostic learning are in the hardest regime when $\text{OPT}_{\mathcal{C}}(D, f)$ goes to $1/2$ as dimension and other problem parameters grow (although there are some notable exceptions in restricted models and the more challenging distribution-independent setting [KS10, FGRW12]). In this work we aim to precisely characterize the value of OPT for which agnostic learning becomes hard and therefore will make this parameter explicit in our lower bounds.

In machine learning literature it is more common to specify the *excess error* which is the difference between $\text{OPT}_{\mathcal{C}}(D, f)$ and the error of the produced hypothesis that an algorithm can achieve. It is easy to see that lower bounds showing that excess error of κ cannot be achieved is equivalent to stating that the lower bound applies to a setting where $\text{OPT} = 1/2 - \kappa$ (since error of $1/2$ can always be achieved).

1.1 Approximate resilience and agnostic learning

In this work we explain why the polynomial ℓ_1 regression algorithm is the best approach known to date for agnostically learning over product distributions. Specifically, we prove that the complexity of agnostic learning \mathcal{C} over a product distribution in the statistical query model is characterized by how well \mathcal{C} can be approximated in the ℓ_1 norm by low-degree polynomials over the same distribution. The statistical query (SQ) model [Kea98] is a well-studied restriction of the PAC learning model in which the learner relies on approximate expectations of functions of an example rather than examples themselves. With the exception of Gaussian elimination¹ all known techniques used in the theory and practice of machine learning have statistical query analogues. Polynomial ℓ_1 regression is no exception, and therefore to prove our characterization it suffices to establish a lower bound on learning by statistical query algorithms for function classes that are not well-approximated by low-degree polynomials.

The optimality of ℓ_1 regression for agnostic learning over product distributions that we prove is based on a formal connection between agnostic learning and a basic structural property of Boolean functions. We say that a function $g : \{-1, 1\}^n \rightarrow \mathbb{R}$ is d -resilient if $\widehat{g}(S) = 0$ for all $|S| \leq d$, i.e. g is uncorrelated with every low-degree parity. Equivalently, g is d -resilient if and only if $\mathbb{E}[g_\rho] = \mathbb{E}[g]$ for any restriction ρ to at most d out of n variables and $\mathbb{E}[g] = 0$. Functions which satisfy the first property are called *correlation*

¹Note that Gaussian elimination fails in the presence of even minor amounts of random noise and is not applicable in the agnostic framework.

immune and are widely-studied for cryptographic applications. The structural question we will be interested in is:

How close can a Boolean function be to a highly resilient function with range in $[-1, 1]$?

More precisely, we say that $f : \{-1, 1\}^n \rightarrow [-1, 1]$ is α -approximately d -resilient if there exists a d -resilient $g : \{-1, 1\}^n \rightarrow [-1, 1]$ such that $\|f - g\|_1 = \mathbf{E}[|f(\mathbf{x}) - g(\mathbf{x})|] \leq \alpha$, and we will be interested in functions that are α -approximately d -resilient for small values of α and large values of d . We note that for simplicity and convenience the definitions here are for the uniform distribution on the hypercube but can be easily extended to general product distributions over other n -dimensional domains (see Section A).

The notion of resilience is well-studied and has applications in cryptography, pseudorandomness, inapproximability, circuit complexity and more (for a few examples, see [CGH⁺85, LW95, AM09, AH11, She11]). However, to the best of our knowledge our notion of approximate resilience does not appear to have been explicitly studied before.

At a high level we show that if a concept class \mathcal{C} contains an α -approximately d -resilient function then the complexity of learning \mathcal{C} agnostically in the SQ model is $n^{\Omega(d)}$. Further, learning is hard even for $\text{OPT} \leq \alpha/2$ (in other words when noise rate is $\alpha/2$). For simplicity the complexity of an SQ algorithm refers to a polynomial upper-bounding both the running time and the inverse of query tolerance. Naturally, the presence of a single α -approximately d -resilient function would not suffice for a hardness result since a concept class with a single function can be easily learned agnostically. We therefore need some assumptions under which existence of a single α -approximately d -resilient function will imply that there are many of them. One such assumption that we adopt is that the α -approximately d -resilient function c depends on at most $n^{1/3}$ variables (such a function is called a $n^{1/3}$ -junta) and the concept class \mathcal{C} is closed under renaming of variables. Alternatively, if we consider an ensemble of concept classes $\{\mathcal{C}_n\}_{n=1}^\infty$ parameterized by dimension n it would be sufficient to assume that the ensemble is closed under addition of irrelevant variables. For brevity we omit the closed-ness under renaming since it is satisfied by all commonly-studied concept classes. We now state our lower bound in terms of resilience informally.

Theorem 1.1. *Let \mathcal{C} be a concept class. Fix d and let $\alpha(d)$ be such that, there exists a $\alpha(d)$ -approximately d -resilient $n^{1/3}$ -junta $c \in \mathcal{C}$. Then any SQ algorithm for agnostically learning \mathcal{C} with excess error of at most $\frac{1-\alpha(d)}{2} - n^{-o(d)}$ has complexity of at least $n^{\Omega(d)}$.*

Alternatively, this result can be stated as saying that if for every function f satisfying $\text{OPT}_{\mathcal{C}}(D, f) \leq \alpha(d)/2$ the algorithm outputs h such that $\Pr_D[h(\mathbf{x}) \neq f(\mathbf{x})] \leq 1/2 - n^{-o(d)}$ then its SQ complexity is $n^{\Omega(d)}$. An immediate implication of this theorem is that a concept class containing an $o(1)$ -approximately d -resilient function cannot be learned with noise rate larger than $o(1)$ in time $n^{\Omega(d)}$.

The proof of this theorem is based on the simple observation that agnostic learning of \mathcal{C} is at least as hard as weak learning of a class of d -resilient functions which are close to functions in \mathcal{C} . From there we rely on hardness of SQ learning of pairwise nearly orthogonal functions to obtain the claim. This result relies crucially on the distribution being a product distribution and it was recently demonstrated that it does not hold for some non-product distributions [FK14].

The lower bounds obtained from this technique are closest in spirit to lower bounds based on cryptographic assumptions and those based on hardness of learning sparse parities with noise. Cryptographic hardness relies on a certain problem being hard for all known “attacks”. As pointed out above, SQ algorithms capture all known agnostic learning algorithms and learning techniques in general. Therefore the lower bounds hold against all known learning algorithms. Further, as in our lower bounds, degree of resilience of a predicate is the primary hardness parameter in many cryptographic constructions (cf. [OW14]).

This simple technique might appear to be a relatively limited approach to obtaining lower bounds. Yet, it turns out that the lower bounds it achieves are essentially optimal. This follows from the duality between

approximate resilience and ℓ_1 approximation by low-degree polynomials that we establish. More formally, let \mathcal{P}_d be the class of degree at most d real-valued polynomials. For a Boolean function f , let $\Delta_{\mathcal{P}_d}(f) = \min_{p \in \mathcal{P}_d} \mathbf{E}[|f - p|]$.

Theorem 1.2. *For $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and $0 \leq d \leq n$ and $\alpha \geq 0$, f is α -approximately d -resilient if and only if $\Delta_{\mathcal{P}_d}(f) \geq 1 - \alpha$.*

The proof of this result is a fairly simple application of a classical result on duality of norms by Ioffe and Tikhomirov [IT68].

Now for a concept class \mathcal{C} , let $\Delta_{\mathcal{P}_d}(\mathcal{C}) = \max_{f \in \mathcal{C}} \Delta_{\mathcal{P}_d}(f)$. To see how this quantity characterizes agnostic learning in the statistical query model, we state the error and running time achieved by the polynomial ℓ_1 regression algorithm of Kalai et al. for agnostic learning [KKMS08]. This algorithm is easy to implement in the SQ model².

Theorem 1.3 ([KKMS08]). *Let \mathcal{C} be a concept class over $\{-1, 1\}^n$ and fix d . There exists a SQ algorithm which for any $\varepsilon > 0$ agnostically learns \mathcal{C} with excess error $\Delta_{\mathcal{P}_d}(\mathcal{C})/2 + \varepsilon$ and has complexity $\text{poly}(n^d, 1/\varepsilon)$.*

On the other hand, we may apply Theorems 1.2 and 1.1 to show that this is the best any SQ algorithm can do; by Theorem 1.2 there exists an $\alpha(d)$ -approximately d -resilient function in \mathcal{C} with $1 - \alpha(d) = \Delta_{\mathcal{P}_d}(\mathcal{C})$. Therefore Theorem 1.1 essentially matches the upper bound of Theorem 1.3 in excess error and complexity, implying the optimality of ℓ_1 -regression based algorithms for agnostic learning over the uniform distribution. The extension to other product distributions is fairly straightforward and we discuss it in Sec. A.

1.2 Learning monotone juntas

With this characterization in hand, we would like to better understand what classes of functions we can hope to agnostically learn on the uniform distribution. Uniform distribution learning is challenging even in the noiseless setting, with efficient algorithms out of reach for natural classes such as polynomial size DNF formulas and decision trees. However, learning monotone functions and their corresponding subclasses seems significantly easier; for example, monotone decision trees [OS07] and monotone DNFs with few terms [Ser01] are efficiently learnable in the SQ model (for other examples see [OW13, BBL98, BT96]).

This difference is demonstrated most dramatically in the junta learning problem, which is considered by many to be the single most important open problem in uniform distribution learning. In this problem, the target function is an unknown k -junta, a Boolean function which depends on at most $k \ll n$ variables. The junta problem also lies at the heart of the notorious DNF and decision tree learning problems: Since s -term DNFs and s -leaf decision trees can compute arbitrary $(\log s)$ -juntas, learning either of these classes requires that we first be able to efficiently learn $\omega(1)$ -juntas. Progress has remained slow in the 20 years since Blum posed the junta problem, with the current fastest algorithm running in time $n^{.60k}$ [Val12], improving on the first non-trivial algorithm which runs in time $n^{.704k}$ [MOS04] (the trivial algorithm exhaustively checks all k -subsets of $[n]$ and runs in time $O(n^k)$). In contrast, monotone juntas are easy to learn using an extremely simple algorithm: the relevant variables can be identified by estimating their correlations with the target function $\mathbf{E}[f(x)x_i] = \hat{f}(\{i\})$, and thus monotone k -juntas can be learned in time $O(n + 2^k)$. Does the advantage of monotonicity hold in the agnostic setting as well? We first consider the simplest problem of agnostic learning monotone juntas. While it appears to be a hard problem, known hardness results for specific monotone functions do not rule out polynomial time algorithms for any constant ε . Specifically, the best known lower bound is $n^{\Omega(1/\varepsilon^2)}$ for majority functions [KKMS08] and is based on the assumption that

²To the best of our knowledge this is not proved anywhere explicitly but is fairly well-known and used in some other works [?]. It follows from the fact that LPs can be optimized approximately using approximate evaluations of the optimized function (in our case expected ℓ_1 error) for example via the Ellipsoid algorithm [Lov87]. See [FPV13] for more details on this general technique.

learning sparse noisy parities is hard. Further, this hardness result only applies when $\text{OPT} \geq 1/2 - \varepsilon$ which leaves open the possibility that the problem is solvable efficiently when the noise rate is a constant smaller than $1/2$.

As we saw in Theorem 1.1, the complexity of agnostic learning of \mathcal{C} is characterized by the approximate resilience of functions in \mathcal{C} . Therefore we consider the structural question of how close monotone functions are to bounded resilient functions. The structure of monotone functions over the Boolean hypercube has been investigated in many influential works (see [BBL98, BT96, MO02, O'D03, OW13]). While to the best of our knowledge our notion has not been studied before, several works have examined the total spectral weight that monotone functions have on low-degree coefficients [BT96, MO02]. Spectral weight indicates the distance to the closest (not necessarily bounded) resilient function in ℓ_2 norm. Both differences of bounded/unbounded and ℓ_1/ℓ_2 are significant, but we show how bounds on low-degree spectral weight can serve as a basis for bounds on our notion of distance to resilience (see Thm. 3.2).

It is easy to see that monotone functions cannot be 1-resilient, and prior to our work, it was possible that every monotone function was $\Omega(1)$ -far from 1-resilient. Our first structural result rules out this possibility in a very strong way:

Theorem 1.4. *For every $\alpha > 0$ there exists an α -approximately d -resilient monotone Boolean function where $d = \Omega(\alpha\sqrt{n}/\log n)$.*

Our proof of this result is indirect and relies crucially on the duality of approximate resilience and ℓ_1 -approximation of monotone functions by polynomials. We use a lower bound for PAC learning of monotone functions by Blum et al. [BBL98] to obtain strong lower bounds on ℓ_1 -approximation of monotone functions by polynomials. We can then use Theorem 1.2 to obtain bounds on distance to resilience.

This degree of resilience is essentially optimal: combining basic facts from discrete Fourier analysis, it is straightforward to see that every monotone Boolean function is α -far from any $\Omega(\alpha\sqrt{n})$ -resilient function [BT96]. Applying our connection between approximate resilience and agnostic learning, we get as a corollary our main application:

Corollary 1.5. *Any SQ algorithm for agnostically learning the class of monotone k -juntas with excess error of $1/2 - \alpha$ has complexity of $n^{\Omega(\alpha\sqrt{k}/\log k)}$.*

Qualitatively, Corollary 1.5 gives the first super-polynomial lower bound on the complexity of SQ algorithms for agnostically learning monotone k -juntas with constant (and even sub-constant) noise. It also rules out the possibility of efficient SQ algorithms for agnostic learning monotone decision trees and monotone DNFs with few terms (which, as previously mentioned, do have efficient SQ algorithms in the noiseless setting). Quantitatively, our lower bound essentially matches the upper bound of $n^{O(\sqrt{k}/\varepsilon)}$ that follows as a corollary of the low-degree concentration bound of [BT96] and the polynomial ℓ_1 regression algorithm [KKMS08]. Note that lower bounds on PAC learning of monotone functions [BBL98] cannot be translated directly to lower bounds in the junta learning setting since these lower bounds are subexponential in k while junta learning algorithms are allowed to run in time polynomial in 2^k .

While Theorem 1.4 yields a near-optimal lower bound on the complexity of agnostically learning general monotone juntas, the construction is not explicit: it is based on a randomized DNF construction (similar to Talagrand's randomized DNF construction [Tal96]), and contains functions of high complexity. Furthermore, for more general classes such as monotone DNFs, the hardness results implied are not optimal. We first show that even the simple Tribes function, a read-once DNF, is close to a resilient function (which gives a stronger hardness result for learning small monotone DNFs).

Theorem 1.6. *Tribes is α -approximately d -resilient, where $\alpha = O(n^{-1/3})$ and $d = \Omega(\log n / \log \log n)$.*

Our proof of Theorem 1.6 is based on a general technique for obtaining bounds on approximate resilience from bounds on spectral weight on low-degree coefficients. Roughly, our result states that for

a sufficiently small γ , if the total spectral weight on degree $\leq d$ coefficients of f is at most γ , then f is $\approx \sqrt{\gamma}e^d$ -approximately d -resilient (see Thm. 3.2). The proof relies on a concentration inequality for low-degree polynomials over independent Rademacher random variables that follows from the hypercontractivity inequalities of Bonami and Beckner [Bon70, Bec75].

We then describe a general technique for amplifying the degree of approximate resilience of functions via iterative composition and apply it to Tribes to obtain an explicit function that is $o(1)$ -approximately $2^{\Omega(\sqrt{\log n})}$ -resilient (see Section 3.4 for details).

Both Theorems 1.4 and 1.6 give monotone Boolean functions which are close to resilient functions, however the resilient functions are not necessarily Boolean-valued. In most cryptographic applications resilience is studied specifically for Boolean functions (e.g., [Sie84, MOS04, OW14]), and therefore it is natural to ask if there are such functions that are close to monotone Boolean functions. Using a new function called CycleRun [Wie], we show that this is indeed possible, and furthermore we nearly match the resilience of the iterated Tribes construction:

Theorem 1.7. *There is an explicit α -approximately d -resilient monotone Boolean function f where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n}/\log \log n)}$. Furthermore, f is α -close to a Boolean d -resilient function.*

We prove Theorem 1.7 by first showing that CycleRun is $O(\sqrt{\log n/n})$ -approximately 1-resilient, where our witness to this approximate resilience is a Boolean function. Our argument crucially relies on four key properties of CycleRun: monotonicity, low influence, oddness, and invariance under cyclic shifts; as far as we know, CycleRun is the only explicit Boolean function known to have all four properties. These properties allow us to use a structured combinatorial argument, unlike our argument for Tribes that relies on properties of polynomials and produces a witness that is a bounded function (and applying this style of argument to Tribes quickly gets unruly). Having established $O(\sqrt{\log n/n})$ -approximate 1-resilience, we then apply the aforementioned general amplification technique to increase the degree of resilience to $2^{\tilde{\Omega}(\sqrt{\log n})}$.

We remark that while the degrees of resilience obtained in Theorems 1.7 and 1.6 are not as strong as that of Theorem 1.4, both are sufficient to rule out the existence of efficient SQ algorithms for learning monotone k -juntas for any $k = \omega_n(1)$ and subconstant error-rate.

1.3 Related work

Lower bounds for statistical query algorithms were first shown by Kearns [Kea98] who proved that parities cannot be learned by SQ algorithms. Soon after this Blum et al. [BFJ⁺94] characterized the weak PAC learnability of every function class \mathcal{C} in the SQ model in terms of the *statistical query dimension* of \mathcal{C} ; roughly speaking, this is the largest number of functions from \mathcal{C} that are pairwise nearly orthogonal to each other (we give a precise definition in Section 2). These lower bound techniques were extended to strong PAC learning and agnostic learning in more recent work [Sim07, Fel12, Szö09]. Lower bounds for SQ algorithms were proved for many learning problems including, for example, PAC learning of juntas [BFJ⁺94], weak-learning of intersections of halfspaces [KS07] and learning of monotone depth-3 formulas [FLS11]. These lower bounds are information-theoretic but capture remarkably well the computational hardness of learning problems. In some cases, such as learning juntas over the uniform distribution, this is the only known formal evidence of the hardness of the problem.

Given the lack of general lower bounds for several basic problems in agnostic learning, many works concentrate on lower bounds against specific popular algorithms such as ℓ_1 -regression [KS10] and margin-based linear methods [LS11, BDLSS12, DLSS14]. These techniques are captured by SQ algorithms and therefore our lower bounds are substantially more general.

Several previously known lower bounds for agnostic learning are based on the reduction to learning of k -sparse noisy parities. This is a notoriously hard problem for which the only non-trivial algorithm is the recent breakthrough result of Valiant that gives an algorithm running in time $n^{0.8k}$ [Val12]. Assuming

that this problem requires $n^{\Omega(k)}$ time we get that agnostic learning of majorities on the uniform distribution requires $n^{\Omega(1/\varepsilon^2)}$ time [KKMS08] and conjunctions require $n^{\Omega(\log(1/\varepsilon))}$ time [Fel12]. Learning k -sparse parities in the SQ model has complexity of $n^{\Omega(k)}$ and therefore these results also give unconditional SQ lower bounds. These lower bounds can be interpreted as special cases of our approach. They are based on showing that a parity of high-degree has a significant correlation with a function in \mathcal{C} . Clearly a k -sparse parity function is $(k - 1)$ -resilient and correlation implies that distance to that parity is slightly better than the trivial 1. The main limitation of this approach is that in most cases it can only lead to hardness results when the noise rate is close to $1/2$. In particular this approach cannot lead to the strong hardness results we prove here for monotone juntas.

In a recent work Feldman and Kothari [FK14] show that the equivalence between ℓ_1 approximation by polynomials and agnostic learning does not extend to non-product distributions. They exhibit a distribution D for which any polynomial that is $1/3$ -close to the disjunction of all the variables in ℓ_1 (measured relative to D) must have degree $\Omega(\sqrt{n})$. At the same time disjunctions are SQ learnable in time $n^{O(\log(1/\varepsilon))}$ over that distribution.

Our approach to proving lower bounds is closest in spirit and shares technical elements with the influential pattern matrix method of Sherstov [She11]. His method shows that lower bounds on the approximation by polynomials in ℓ_∞ norm of a function f can be translated into lower bounds on randomized communication complexity of a certain communication problem corresponding to evaluation of f on different subsets of variables (which were previously thought as stronger than lower bounds on approximation in ℓ_∞ by polynomials). A crucial step in his result is an application of duality that is in some sense symmetric to ours and shows the existence of an unbounded resilient function g that is correlated with f . Such g then serves to upper bound discrepancy for the communication problem (from which a lower bound on randomized communication complexity follows).

1.4 Preliminaries

All probabilities and expectations are with respect to the uniform distribution unless otherwise stated, and we will use boldface (e.g. \mathbf{x} and \mathbf{y}) to denote random variables. Given $f, g : \{-1, 1\}^n \rightarrow \mathbb{R}$, we say that f and g are ε -close if $\|f - g\|_1 = \mathbf{E}[|f(\mathbf{x}) - g(\mathbf{x})|] \leq \varepsilon$. We say that g is bounded if it takes values in the interval $[-1, 1]$. Note that if f is Boolean valued and g is bounded, then $\|f - g\|_1 = 1 - \mathbf{E}[fg]$. Every function $g : \{-1, 1\}^n \rightarrow \mathbb{R}$ can be uniquely written as a multilinear polynomial such that $g(\mathbf{x}) = \sum_{S \subseteq [n]} \widehat{g}(S) \prod_{i \in S} x_i$

for all $\mathbf{x} \in \{-1, 1\}^n$; the coefficients $\widehat{g}(S)$ are called the Fourier coefficients of g . The total influence of a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$, denoted $\text{Inf}[f]$, is $\sum_{i=1}^n \Pr[f(\mathbf{x}) \neq f(\mathbf{x}^{\oplus i})]$, where $\mathbf{x}^{\oplus i}$ denotes \mathbf{x} with its i -th coordinate flipped.

Definition 1.8. A function $g : \{-1, 1\}^n \rightarrow \mathbb{R}$ is d -resilient if $\widehat{g}(S) = 0$ for all $|S| \leq d$. We say that a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is α -approximately d -resilient if there exists a d -resilient bounded function g such that $\|f - g\|_1 \leq \alpha$.

Learning background In the agnostic learning framework, the learning algorithm is given labeled examples (\mathbf{x}, \mathbf{y}) where $\mathbf{x} \in \{-1, 1\}^n$ and $\mathbf{y} \in \{-1, 1\}$ are drawn from a distribution \mathcal{D} over $\{-1, 1\}^n \times \{-1, 1\}$. As usual we describe such distributions by a pair (D, g) , where D is the marginal distribution on $\{-1, 1\}^n$ and $g : \{-1, 1\}^n \rightarrow [-1, 1]$, where $g(\mathbf{x}) = \mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[\mathbf{y} \mid \mathbf{x} = \mathbf{x}]$ is expectation of the label for each input. Note that for every Boolean function f , if U denotes the uniform distribution then $\mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim (U, g)}[f(\mathbf{x}) \neq \mathbf{y}] = \|f - g\|_1/2$.

Definition 1.9. Let \mathcal{C} be a class of Boolean functions on $\{-1, 1\}^n$. An algorithm A agnostically learns \mathcal{C} over distribution D on $\{-1, 1\}^n$ if for any $g : \{-1, 1\}^n \rightarrow [-1, 1]$ and $\varepsilon > 0$, given examples from

distribution $\mathcal{D} = (D, g)$ and ε , it outputs with probability at least $2/3$ hypothesis $h : \{-1, 1\}^n \rightarrow \{-1, 1\}$ such that:

$$\Pr[h(\mathbf{x}) \neq \mathbf{y}] \leq \text{OPT}_{\mathcal{C}}(D, g) + \varepsilon,$$

where $\text{OPT} = \min_{c \in \mathcal{C}} \Pr_{(\mathbf{x}, \mathbf{y}) \sim (D, g)}[c(\mathbf{x}) \neq \mathbf{y}]$. The algorithm is said to learn with excess error κ if h instead satisfies

$$\Pr[h(\mathbf{x}) \neq \mathbf{y}] \leq \text{OPT}_{\mathcal{C}}(D, g) + \kappa.$$

Definition 1.10. A statistical query is defined by a bounded function of an example $\phi : \{-1, 1\}^n \times \{-1, 1\} \rightarrow [-1, 1]$ and positive tolerance τ . A valid reply to such a query relative to a distribution \mathcal{D} over examples is a value v that satisfies:

$$|\mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[\phi(\mathbf{x}, \mathbf{y})] - v| \leq \tau.$$

A statistical query learning algorithm is an algorithm which relies solely on statistical queries and does not have access to actual examples. We say that an SQ algorithm has **statistical query complexity** T if it makes at most q statistical queries of tolerance at least τ and $T \geq \max\{q, 1/\tau\}$.

2 Characterization of Agnostic Learning

In this section we show that approximate resilience implies hardness of agnostic learning for statistical query algorithms (Lemma 2.1). We then show that the implication works in the reverse direction as well: if a class does not contain approximately resilient functions, then it can be agnostically learned by SQ algorithms. We prove this equivalence using the duality between approximate resilience and approximation by low-degree polynomials stated in Theorem 1.2. This simple observation turns out to be surprisingly useful, leading both to a characterization of agnostic learning and to a proof of our first structural result for monotone functions (Theorem 1.4).

To connect our notion of approximate resilience to the hardness of agnostic learning we will use the following standard notion of designs of sets with small overlap. A (n, k, d) -design of size m is a collection of sets $S_1, \dots, S_m \subseteq [n]$ such that $|S_i| = k$ and $|S_i \cap S_j| \leq d$ for all $i \neq j$. Let $\mathcal{M}(n, k, d)$ denote the size of the largest (n, k, d) -design. Standard probabilistic/greedy argument implies that

$$\mathcal{M}(n, k, d) \geq \frac{\binom{n}{k}}{\binom{k}{d} \binom{n-d}{k-d}} = \frac{\binom{n}{d}}{\binom{k}{d}^2} \geq \left(\frac{nd}{e^2 k^2} \right)^d. \quad (1)$$

For a function $f : \{-1, 1\}^k \rightarrow \{-1, 1\}$ and set $S \subseteq [n]$ of size k we use $f_S : \{-1, 1\}^n \rightarrow \{-1, 1\}$ to denote $f(\mathbf{x}_{|S})$ where $\mathbf{x}_{|S}$ refers to the restriction of \mathbf{x} to coordinates with indices in S (in the usual order).

Lemma 2.1. Let $f : \{-1, 1\}^k \rightarrow \{-1, 1\}$ be an α -approximately d -resilient function. Let S_1, \dots, S_m be a (n, k, d) -design. If $\{f_{S_i}\}_{i=1}^m \subseteq \mathcal{C}$, then any SQ algorithm for agnostically learning \mathcal{C} with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$ has complexity of at least $m^{1/3}$.

To prove Lemma 2.1, we will use the following result implicit in [Fel12] that is a simple generalization of the well-known SQ-DIM bounds from [BFJ⁺94] and their strengthening in [Yan05, Szö09].

Theorem 2.2. Let D be a distribution and let g_1, \dots, g_m be bounded real-valued functions such that $|\langle g_i, g_j \rangle_D| \leq 1/m$ for $i \neq j$, where $\langle g_i, g_j \rangle_D = \mathbf{E}_D[g_i(\mathbf{x}) \cdot g_j(\mathbf{x})]$. Then any SQ algorithm that for every i , given access to statistical queries with respect to distribution (D, g_i) outputs a hypothesis h such that $\mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim (D, g_i)}[h(\mathbf{x}) \neq \mathbf{y}] \leq \frac{1}{2} - \frac{1}{m^{1/3}}$ has complexity of at least $m^{1/3}$.

We can now prove Lemma 2.1.

Proof. By our assumption, the function f is α -close to a d -resilient bounded function $g : \{-1, 1\}^k \rightarrow [-1, 1]$. We first note that each pair of functions g_{S_i}, g_{S_j} shares at most d relevant variables. These functions are d -resilient and therefore there is no single set T such that $\widehat{g_{S_i}}(T) \cdot \widehat{g_{S_j}}(T) \neq 0$. This, by linearity of expectation implies that for $i \neq j$, $\mathbf{E}[g_{S_i} g_{S_j}] = 0$.

Let A be an agnostic algorithm for \mathcal{C} with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$. For every i , f_{S_i} is α -close to g_{S_i} . Therefore if the input distribution is (U, g_i) then $\text{OPT}_{\mathcal{C}}(U, g_i) \leq \|f_{S_i} - g_{S_i}\|_1/2 = \|f - g\|_1/2 \leq \alpha/2$. This implies that A will output a hypothesis h with error of at most $\alpha/2 + \frac{1-\alpha}{2} - m^{-1/3} = 1/2 - m^{-1/3}$. By Theorem 2.2 and orthogonality of g_{S_i} s we get that the complexity of A is at least $m^{1/3}$. \square

An immediate corollary of Lemma 2.1 is the following lower bound that generalizes Theorem 1.1.

Theorem 2.3. *Let \mathcal{C} be a concept class closed under renaming of variables and assume that \mathcal{C} contains an α -approximately d -resilient k -junta. Then any SQ algorithm for agnostically learning \mathcal{C} with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$ has complexity of at least $m^{1/3}$, where $m = \mathcal{M}(n, k, d)$. In particular, for any constant $\delta > 0$ and $k = n^{1/2+\delta}$, we have $m = n^{\Omega(d)}$.*

To show that Theorem 2.3 is essentially tight we prove the duality stated in Theorem 1.2 (which we restate here for convenience).

Theorem. [Thm. 1.2 restated] *For $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and $0 \leq d \leq n$ let α denote the ℓ_1 distance of f to the closest d -resilient bounded function. Then $\Delta_{\mathcal{P}_d}(f) = 1 - \alpha$.*

Proof. Our proof is an adaptation of the general results on duality of norms [IT68] to the case where f is Boolean and g is bounded. In this case it is easy to see that $\|f - g\|_1 = 1 - \mathbf{E}[fg]$ and therefore minimization of distance to resilience can be expressed as maximization of $\sum_x f(x)g(x)$ subject to resilience constraints on g . Viewing values of $g(x)$ as variables we get:

$$\begin{aligned} & \max \sum_x f(x)g(x) \\ & \text{subject to } \sum_x g(x)\chi_S(x) = 0 \quad \forall |S| \leq d \\ & \text{and } |g(x)| \leq 1 \quad \forall x \in \{-1, 1\}^n \end{aligned}$$

The dual LP can be easily verified to be the following program with variables p_S for every $S \subseteq [n]$ of size at most d .

$$\begin{aligned} & \min \sum_x |q(x)| \\ & \text{subject to } q(x) = f(x) - \sum_{S: |S| \leq d} p_S \chi_S(x) \quad \forall x \in \{-1, 1\}^n \end{aligned}$$

Now the claim of the theorem follows from LP duality. By definition the maximum value of the primal is $2^n \cdot \mathbf{E}[fg] = 2^n(1 - \|f - g\|_1) = 2^n(1 - \alpha)$. This is therefore also the minimum of the dual program which, by definition, is exactly $2^n \cdot \Delta_{\mathcal{P}_d}(f)$. \square

Note that $(1 - \alpha)/2$ in the excess error term in the statement of Theorem 2.3 is equal to $\Delta_{\mathcal{P}_d}(\mathcal{C})/2$ in the excess error term in the statement Theorem 1.3. Therefore combining the duality with the upper-bounds on polynomial ℓ_1 regression stated in Theorem 1.3 we get our claimed characterization of the complexity of agnostic learning in terms of $\Delta_{\mathcal{P}_d}(\mathcal{C})$ or, alternatively, distance to d -resilience.

3 Monotonicity and approximate resilience

In this section we prove bounds on the approximate resilience of monotone functions. First, we give a bound for general monotone functions (Theorem 1.4) in Section 3.1. In Sections 3.2 and 3.3 we show that Tribes and CycleRun are approximately resilient (Theorems 1.6 and 1.7). Finally, in Section 3.4 we show how these functions can be used in an iterated construction to yield explicit functions with high approximate resilience.

3.1 A monotone function with nearly-optimal approximate resilience

Our characterization suggests an approach for proving Theorem 1.4: since the ℓ_1 -minimization algorithm characterizes SQ agnostic learning, we seek monotone functions where the ℓ_1 -minimization algorithm will badly fail. In other words, our first step will be to move to the dual problem: Theorem 1.2 tells us that we may equivalently show the existence of a monotone function f which is far from every low-degree polynomial p . Strangely, to show that no dual solution exists, we will use the fact that if every monotone function had a weak approximation by some low-degree polynomial, then the ℓ_1 -minimization algorithm would learn monotone functions, contradicting known information-theoretic lower bounds [BBL98]. Note that while the ℓ_1 -minimization algorithm is presented as an agnostic learning algorithm, we may apply it directly to the class of monotone functions.

We now prove Theorem 1.4:

Theorem. *For every $\alpha > 0$, there is a monotone function that is α -approximately d -resilient for $d = \Omega(\alpha\sqrt{n}/\log n)$.*

Proof. We show the existence of a monotone function f such that $\mathbf{E}[|f(\mathbf{x}) - p(\mathbf{x})|] > 1 - \alpha$ for every degree- d polynomial p and then apply Theorem 1.2. Suppose that every monotone f satisfies $\mathbf{E}[|f(\mathbf{x}) - p(\mathbf{x})|] \leq 1 - \alpha$. Then for $\varepsilon = \alpha/4$, Theorem 1.3 gives an algorithm for learning monotone functions which uses $s = \text{poly}(n^d/\alpha)$ examples and has error $1/2 - \alpha/2 + \alpha/4 = 1/2 - \alpha/4$. We now use an information-theoretic lower bound on the number of random examples needed to weakly learn monotone functions; the proof in [BBL98] uses a randomized construction of DNF formulas:

Theorem 3.1 ([BBL98]). *Let A be any learning algorithm that uses s random examples and outputs a hypothesis h . Then there is some monotone $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ such that*

$$\Pr[f(\mathbf{x}) = h(\mathbf{x})] \leq \frac{1}{2} + O\left(\frac{\log sn}{\sqrt{n}}\right).$$

Theorem 3.1 tells us that $\alpha = O\left(\frac{d \log n + \log 1/\alpha}{\sqrt{n}}\right)$, which completes the proof. \square

The function from Theorem 1.4 gives us a k -junta that is α -approximately d -resilient for $d = \Omega(\alpha\sqrt{k}/\log k)$. Plugging this into Theorem 2.3 and using eq.(1) (assuming $k \leq n^{1/2}$) we obtain the proof of Corollary 1.5.

While the degree of resilience in Theorem 1.4 is nearly optimal, the proof is non-constructive and relies crucially on the fact that monotone functions can have high complexity. In the following sections we show that even simple, explicit monotone functions can exhibit high approximate resilience.

3.2 Tribes is approximately resilient

The $\text{Tribes}_{w,s} : \{-1, 1\}^{sw} \rightarrow \{-1, 1\}$ function is the disjunction of s disjoint monotone conjunctions, each of width w ; i.e. a read-once width- w DNF. For notational brevity we write Tribes to denote $\text{Tribes}_{w,s}$ with $s = (\ln 2)2^w$ (so $w \approx \log n - \log \ln n$ and $s \approx n/(\log n)$).

Our construction of a highly resilient function close to Tribes is based on a general result relating the low-degree Fourier weight of a Boolean function and its approximate resilience.

Theorem 3.2. *There exists a universal $K > 0$ such that the following holds. Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a Boolean function that satisfies $\sum_{|S| \leq d} \widehat{f}(S)^2 \leq \gamma$ for some $d \in [n]$ and $\gamma \in [0, 1]$. Then for all $\tau > e^d \sqrt{\gamma}$, we have that f is $O(\tau + \delta n^{2d+2})$ -approximately d -resilient, where $\delta = \exp(-K(\tau^2/\gamma)^{1/d})$.*

We now prove Theorem 3.2, and in Section 3.2.1 we show how Theorem 1.6 (i.e. the approximate resilience of Tribes) follows as a consequence of Theorem 3.2.

We begin our construction with the Fourier polynomial for f and discard the low-degree terms. That we may do so and hope to arrive at a bounded, resilient function comes from hypercontractivity: since the discarded polynomial has low-degree, it will be highly concentrated around its mean. The following Chernoff-type concentration inequality for low-degree polynomials over independent Rademacher random variables follows from the hypercontractivity inequalities of Bonami and Beckner [Bon70, Bec75] (see for example [O'D13]).

Theorem 3.3 (concentration of degree- d polynomials). *There exists a universal constant $K > 0$ such that for every degree- d polynomial $\{-1, 1\}^n \rightarrow \mathbb{R}$ and $t > e^d$, we have*

$$\Pr_x[|p(x)| \geq t \cdot \|p\|_2] \leq \exp(-Kt^{2/d}).$$

We now begin the proof of Theorem 3.2. Let

$$\ell(x) = \sum_{|S| \leq d} \widehat{f}(S) \chi_S(x), \quad \text{and} \quad h(x) = f(x) - \ell(x).$$

Our final resilient, bounded function p will be based on h , the high-degree part of f . Note that while h is d -resilient by definition, it may not be uniformly bounded. However, the degree- d Chernoff bound applied to ℓ (the low-degree part), together with our assumption on the variance of ℓ (i.e. the low-degree Fourier weight of f), tell us that ℓ does not attain large values very often. Therefore, while h may not be uniformly bounded, we have that h is bounded on almost all inputs x since $h(x) + \ell(x) = f(x) \in \{-1, 1\}$.

More formally, we set $t = \tau/\sqrt{\gamma}$ in Theorem 3.3 (since $\tau > e^d \sqrt{\gamma}$, we have that indeed $t > e^d$)

$$\Pr_x[|\ell(x)| \geq \tau] \leq \exp(-K(\tau^2/\gamma)^{1/d}) := \delta.$$

Next, we define $q : \{-1, 1\}^n \rightarrow \mathbb{R}$ to be such that

$$q(x) = \begin{cases} 0 & \text{if } |\ell(x)| > \tau \\ h(x) & \text{if } |\ell(x)| \leq \tau. \end{cases}$$

Since $h(x) = f(x) - \ell(x)$ and f is $\{-1, 1\}$ -valued, the range of q is $[-1 - \tau, 1 + \tau]$. While q is bounded, it may now have correlations with low-degree terms (i.e. q is no longer resilient like h is). However, we may also write q as $q(x) = h(x) - h(x) \cdot \mathbf{1}_{|\ell| > \tau}(x)$, where h is d -resilient and $\mathbf{1}_{|\ell| > \tau}$ has very small support. Thus, we will show that we may discard the low-degree terms of q and the effect on boundedness will be uniformly small.

Let $q_{>d}(x) = \sum_{|S| \geq d+1} \widehat{q}(S) \chi_S(x)$, $q_{\leq d} = q - q_{>d}$ and $p(x) = \frac{q_{>d}(x)}{\|q_{>d}\|_\infty}$. Certainly, the range of p is $[-1, 1]$; it remains to bound the correlation of p with f . We have that:

$$\begin{aligned} \mathbf{E}[p \cdot f] &= \mathbf{E}\left[\frac{(q - q_{\leq d})}{\|q_{>d}\|_\infty} \cdot f\right] \\ &\geq \frac{1}{\|q\|_\infty + \|q_{\leq d}\|_\infty} \cdot (\mathbf{E}[q \cdot f] - \|q_{\leq d}\|_\infty) \end{aligned} \tag{2}$$

The correlation of f with q is large:

$$\mathbf{E}_x[q(x) \cdot f(x)] \geq (1 - \tau)(1 - \delta) \geq 1 - \tau - \delta. \quad (3)$$

The above holds because the contribution to the correlation is 0 when $q(x) = 0$, which happens on at most a δ fraction of the inputs. On the remaining inputs, $q(x) = h(x) = f(x) - \ell(x)$, and we assumed $|\ell(x)| \leq \tau$. Thus the contribution on such x is

$$q(x) \cdot f(x) = (f(x) - \ell(x)) \cdot f(x) = 1 - \ell(x) \cdot f(x) \geq 1 - |\ell(x)| \geq 1 - \tau.$$

Thus, it only remains to bound the maximum value of the low-degree part of q :

Claim 3.4.

$$\|q_{\leq d}\|_{\infty} \leq \delta n^{2d+2}$$

Proof. We will show that $|\widehat{q}(S)| < \delta n^{d+1}$ holds for any $|S| \leq d$. Recalling that $q(x) = h(x) - \mathbf{1}_{|\ell|>\tau} \cdot h(x)$, we have:

$$\begin{aligned} \widehat{q}(S) &= \widehat{h}(S) - \widehat{\mathbf{1}_{|\ell|>\tau} \cdot h}(S) \\ |\widehat{q}(S)| &\leq |\widehat{h}(S)| + \mathbf{E}[\|\mathbf{1}_{|\ell|>\tau} \cdot h\|] \\ &\leq 0 + \delta \cdot \|h\|_{\infty} \\ &\leq \delta(\|\ell\|_{\infty} + 1), \end{aligned}$$

where the second inequality holds when $|S| \leq d$ because h is d -resilient, and the last inequality holds because $|h(x)| \leq |\ell(x)| + 1$ for all x . As f is a Boolean function, each of the non-zero Fourier coefficients of ℓ is at most 1 in magnitude. The rough bound of n^{d+1} on the number of non-zero coefficients of ℓ gives a bound of n^{d+1} on $\|\ell\|_{\infty}$; summing over at most n^{d+1} terms of degree at most d gives the claim. \square

Let $\kappa = \delta n^{2d+2}$. Substituting into Equations (2) and (3), we have that

$$\mathbf{E}_x[p(x) \cdot \text{Tribes}(x)] \geq \frac{1 - \tau - \delta - \kappa}{1 + \tau + \kappa} \geq 1 - \delta - 2\tau - 2\kappa,$$

using the fact that $1/(1+x) \geq 1-x$ for $x \geq 0$, and this completes the proof of Theorem 3.2.

3.2.1 Proof of Theorem 1.6

To apply Theorem 3.2 we will need the following upper bound on the low-degree Fourier weight of Tribes, whose proof is given in Appendix B, can be obtained using the explicit values of each Fourier coefficient given in [Man95],

Proposition 3.5. *For any $d \leq w$ the Fourier weight of Tribes on degree d and below is at most*

$$\sum_{|S| \leq d} \widehat{\text{Tribes}}(S)^2 \leq 2 \frac{(2 \ln n)^{2d+4}}{n}.$$

To derive Theorem 1.6 from Theorem 3.2, we set $\tau = (2 \ln n)^{3d} n^{-2/5}$, so that $t := \tau/\sqrt{\gamma} \geq n^{1/10}$. Now there exists a small constant $c > 0$ such that for $d = c \log n / \log \log n$ and large enough n , we have that $\tau = O(n^{-1/3})$, $t > e^d$ and $t^{2/d} \geq n^{1/(5d)} \geq \frac{3}{K} (\log n)^2 \geq \frac{(2d+3)}{K} \ln n$. This implies that $\delta := \exp(-K t^{2/d}) \leq n^{-2d-3}$ and so $\delta n^{2d+2} \leq 1/n$. We conclude that Tribes is α -approximately d -resilient where $\alpha = O(\tau + n^{-1}) = O(n^{-1/3})$, and this completes the proof of Theorem 1.6.

3.3 CycleRun is approximately resilient: Proof of Theorem 1.7

Definition 3.6. For every n , the CycleRun Boolean function $\text{CycleRun} : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is defined as follows: Call a consecutive sequence of 1's a 1-run. Similarly, a consecutive sequence of -1's is a -1-run. We allow runs to wrap around, so if a run reaches x_n it may continue with x_1 . The value of CycleRun is the winner (1 for 1-player or -1 for -1-player) from the following procedure:

1. Check which player has the longest run.
2. In case of tie check which player has a larger number of maximum-length runs.
3. In case of tie check the total length of segments between maximum-length runs, where a segment starting from a 1-run clockwise is counted for the 1-player and a segment starting at a -1-run clockwise is counted for the -1-player. The player that has a larger total count is declared the winner.

We will need that fact that CycleRun has influence $O(\log n)$. Since the proof of this fact has not appeared in the literature before, we include a proof in Appendix C.1 for completeness.

Theorem 3.7. There exist universal constants c_1, c_2 such that for every $n \geq c_2$, there exists a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ such that:

1. For all $S \subseteq [n]$ such that $|S| \leq 1$, $\widehat{f}(S) = 0$, and
2. $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot \text{CycleRun}(\mathbf{x})] \geq 1 - c_1 \sqrt{(\log n)/n}$.

Our proof of Theorem 3.7 relies on four key properties of CycleRun: monotonicity, low influence, oddness, and invariance under cyclic shifts; as far as we know, CycleRun is the only explicit Boolean function known to have all four properties. First, as CycleRun is monotone and transitive, we note that

$$\widehat{\text{CycleRun}}(\{i\}) = \widehat{\text{CycleRun}}(\{j\}) = O\left(\frac{\log n}{n}\right) \quad \text{for all } i \neq j \in [n].$$

The high level intuition behind our proof is simple: we show that by flipping the values of CycleRun from the top of the hypercube downwards and bottom upwards simultaneously, we obtain a balanced function with no Fourier weight at the first level. This can be done without changing too many points because CycleRun has small influence; we are able to do it in a controlled way because it is additionally odd and invariant under cyclic shifts. We defer the proof of Theorem 3.7 to Appendix C.

It is natural to wonder how close a monotone function can be to a 1-resilient Boolean function. We show in Appendix C.2 that Theorem 3.7 is tight:

Theorem 3.8. For every monotone function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and 1-resilient $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$, we have $\Pr_{\mathbf{x}}[f(\mathbf{x}) \neq g(\mathbf{x})] \geq \Omega\left(\sqrt{\frac{\log n}{n}}\right)$.

3.4 Resilience amplification

In this section we prove a general amplification lemma for resilience. Given a value $t \in [-1, 1]$, we write $\mathbf{b}(t)$ to denote a random ± 1 bit with expected value t :

$$\mathbf{b}(t) = \begin{cases} 1 & \text{with probability } (1+t)/2 \\ -1 & \text{with probability } (1-t)/2. \end{cases}$$

(In particular, $\mathbf{b}(1)$ is the constant 1 and $\mathbf{b}(-1)$ is the constant -1). Given bounded functions $G : \{-1, 1\}^m \rightarrow [-1, 1]$ and $g : \{-1, 1\}^n \rightarrow [-1, 1]$, we define their (disjoint) composition $G \circ g : \{-1, 1\}^{mn} \rightarrow [-1, 1]$ to

be $(G \circ g)(x^1, \dots, x^m) := \mathbf{E}[G(\mathbf{b}(g(x^1)), \dots, \mathbf{b}(g(x^m)))]$. Note that if $\mathbf{E}[g(x)] = 0$, then $\mathbf{E}[\mathbf{b}(g(x))] = 0$ as well. Throughout this section we write $\text{dist}(f, g)$ to denote $\frac{1}{2}\mathbf{E}[|f(x) - g(x)|]$ for notational brevity (this is simply the fractional Hamming distance $\Pr[f(x) \neq g(x)]$ when f and g are $\{\pm 1\}$ -valued).

The main result in this section is the following amplification lemma:

Theorem 3.9. *Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and $g : \{-1, 1\}^n \rightarrow [-1, 1]$ where $\mathbf{E}[f(x)] = \mathbf{E}[g(x)] = 0$, and suppose g is d -resilient. Consider the recursively-defined functions where $f_k = f \circ f_{k-1}$ and $g_k = g \circ g_{k-1}$ for all $k \in \mathbb{N}$, and $f_0 = f$ and $g_0 = g$. Then for $k \geq 1$:*

1. f_k and g_k are functions over n^{k+1} variables,
2. g_k is $((d+1)^{k+1} - 1)$ -resilient,
3. $\text{dist}(f_k, g_k) \leq \text{dist}(f, g) \sum_{t=0}^k \text{Inf}[f]^t$.

The first claim is straightforward to verify, and so we focus on the second and third claims. For a Boolean-valued function $F : \{-1, 1\}^m \rightarrow \{-1, 1\}$ and $\delta \in [0, 1]$, recall that the *noise-sensitivity* of F at noise rate δ is defined as $\text{NS}_\delta[F] := \Pr_{\mathbf{y}, \mathbf{z}}[F(\mathbf{y}) \neq F(\mathbf{z})]$, where \mathbf{y} is uniform in $\{-1, 1\}^m$ and \mathbf{z} is obtained from \mathbf{y} by independently flipping each of its coordinates with probability δ .

Lemma 3.10. *Given $F, f : \{-1, 1\}^m \rightarrow \{-1, 1\}$ and $G, g : \{-1, 1\}^m \rightarrow [-1, 1]$ where $\mathbf{E}[f(x)] = \mathbf{E}[g(x)] = 0$, we have*

$$\text{dist}(F \circ f, G \circ g) \leq \text{dist}(F, G) + \text{NS}_\delta[F],$$

where $\delta := \text{dist}(f, g)$.

Proof. We first apply the triangle inequality and note that

$$\text{dist}(F \circ f, G \circ g) \leq \text{dist}(F \circ f, F \circ g) + \text{dist}(F \circ g, G \circ g).$$

Since $\mathbf{E}[g(x)] = 0$, we have that $\langle \mathbf{b}(g(x^1)), \dots, \mathbf{b}(g(x^m)) \rangle$ is uniformly distributed on $\{-1, 1\}^m$ when x^1, \dots, x^m are independently and uniformly distributed on $\{-1, 1\}^n$, and therefore the second distance on the right hand side is exactly $\text{dist}(F, G)$. Since $\Pr[\mathbf{b}(f(x)) \neq \mathbf{b}(g(x))] = \Pr[f(x) \neq \mathbf{b}(g(x))] = \frac{1}{2}|f(x) - g(x)|$ for all $x \in \{-1, 1\}^n$, it follows that $\Pr[\mathbf{b}(f(x)) \neq \mathbf{b}(g(x))] = \frac{1}{2}\mathbf{E}[|f(x) - g(x)|] = \delta$ and so

$$\text{dist}(F \circ f, F \circ g) = \Pr_{\mathbf{y}, \mathbf{z}}[F(\mathbf{y}) \neq F(\mathbf{z})],$$

where \mathbf{y} is uniform in $\{-1, 1\}^m$ and \mathbf{z} is obtained from \mathbf{y} by independently flipping each of its coordinates with probability δ . This completes the proof, since the probability on the right hand side is precisely $\text{NS}_\delta[F]$. \square

Using the union bound, we have

$$\text{NS}_\delta[F] \leq \delta \sum_{i=1}^n \Pr_{\mathbf{x}}[F(\mathbf{x}) \neq F(\mathbf{x}^{\oplus i})] = \delta \cdot \text{Inf}[F] = \text{dist}(f, g) \cdot \text{Inf}[F],$$

where $\mathbf{x}^{\oplus i}$ is the string \mathbf{x} with the i -th bit flipped, and $\delta = \text{dist}(f, g)$ as in the previous lemma. This, along with a straightforward recursion, yields the following corollary.

Corollary 3.11. *Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and $g : \{-1, 1\}^n \rightarrow [-1, 1]$ where $\mathbf{E}[f(x)] = \mathbf{E}[g(x)] = 0$, and suppose g is d -resilient. Consider the recursively-defined functions where $f_k = f \circ f_{k-1}$ and $g_k = g \circ g_{k-1}$ for all $k \in \mathbb{N}$, and $f_0 = f$ and $g_0 = g$. Then for $k \geq 1$:*

$$\text{dist}(f_k, g_k) \leq \text{dist}(f, g) \sum_{t=0}^k \text{Inf}[f]^t.$$

Lemma 3.12. *If $G : \{-1, 1\}^m \rightarrow [-1, 1]$ is d_1 -resilient and $g : \{-1, 1\}^n \rightarrow [-1, 1]$ is d_2 -resilient, then $G \circ g$ is $(d_1 d_2)$ -resilient.*

Proof. By linearity of the Fourier transform it suffices to prove this claim when $G(x_1, \dots, x_m) = \prod_{i \in T} x_i$ and $|T| > d_1$, the parity function over $d_1 + 1$ or more variables. We begin by noting that

$$\begin{aligned} (G \circ g)(x^1, \dots, x^m) &= \mathbf{E} \left[\prod_{i \in T} b(g(x^i)) \right] \\ &= \prod_{i \in T} \mathbf{E}[b(g(x^i))] \\ &= \prod_{i \in T} \left[\frac{1 + g(x^i)}{2} - \frac{1 - g(x^i)}{2} \right] = \prod_{i \in T} g(x^i). \end{aligned}$$

We view the mn coordinates of the composed function $G \circ g$ as the disjoint union of $A_1 \cup \dots \cup A_m$, where each A_i has size n . With this notation in hand, every subset S of the mn coordinates may be viewed as the disjoint union $S_1 \cup \dots \cup S_m$, where $A_j \subseteq S_j$ for all $j \in [m]$. Fix $S = S_1 \cup \dots \cup S_m$ of cardinality at most $d_1 d_2$, and recall that our goal is to show that $\widehat{(G \circ g)}(S) = 0$. There exists at least one set S_j where $|S_j| \leq d_2$, and we assume without loss of generality that $|S_1| \leq d_2$. Since g is d_2 -resilient (in particular, $\widehat{g}(S_1) = 0$), we see that indeed

$$\widehat{(G \circ g)}(S) = \mathbf{E} \left[\prod_{i \in T} g(x^i) \prod_{j \in [m]} \prod_{\ell \in S_j} x_\ell^j \right] = \prod_{i \in T} \widehat{g}(S_i) \prod_{j \notin T} \prod_{\ell \in S_j} \mathbf{E}[x_\ell^j] = 0,$$

and the proof is complete. \square

Combining Corollary 3.11 and Lemma 3.12 yields Theorem 3.9.

3.4.1 Amplifying Tribes and CycleRun

We now apply Theorem 3.9 to Tribes and CycleRun.

Theorem 3.13. *There is an explicit α -approximately d -resilient monotone Boolean function F where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n})}$.*

Proof. We apply Theorem 3.9 with f being Tribes and g the bounded resilient function that results from applying Theorem 1.6. Since $\text{Inf}[\text{Tribes}] = \Theta(\log n)$ (see e.g. [KKL88]), taking $k := c \log n / \log \log n$ where $c > 0$ is a sufficiently small universal constant gives functions f_k, g_k over $N := n^k = 2^{O(\log^2 n / \log \log n)}$ variables, where

$$\text{dist}(f_k, g_k) = O(\text{Inf}[\text{Tribes}]^{k+1} \cdot n^{-1/3}) = n^{-\Omega(1)} = o_N(1),$$

and g_k is d -resilient for

$$d = \Omega((\log n / \log \log n)^{k+1}) = 2^{\Omega(\sqrt{\log N})}.$$

\square

Analogous calculations for CycleRun yield the following:

Theorem 1.7. *There is an explicit α -approximately d -resilient monotone Boolean function F where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n / \log \log n})}$. Furthermore, F is α -close to a d -resilient function that is Boolean-valued as well.*

Proof. We apply Theorem 3.9 with f being CycleRun and g the Boolean-valued resilient function that results from applying Theorem 3.7. Since $\text{Inf}[\text{CycleRun}] = O(\log n)$ (Theorem C.6), we again take $k = c \log n / \log \log n$ where $c > 0$ is a sufficiently small universal constant to get Boolean-valued functions f_k, g_k over $N = 2^{O(\log^2 n / \log \log n)}$ variables, where $\Pr[f_k(\mathbf{x}) \neq g_k(\mathbf{x})] = \text{dist}(f_k, g_k) = n^{-\Omega(1)} = o_N(1)$, and g_k is d -resilient for $d = n^{\Omega(1/\log \log n)} = 2^{O(\sqrt{\log N} / \log \log N)}$. \square

4 Conclusions

We have demonstrated that complexity of agnostic learning over product distributions has a natural characterization via either of two dual notions: ℓ_1 -approximation by polynomials and approximate resilience. The notion of distance to resilience that we introduce appears to be interesting its own right. It is also better suited for proving lower bounds since a single close resilient function witnesses the hardness of agnostic learning. Our proof of this result is relatively simple and remarkably, up to the choice of norms, is identical to Sherstov’s powerful pattern matrix method in communication complexity [She11].

An application of our characterization and our second contribution is new and detailed picture of the hardness of agnostic learning of monotone functions over the uniform distribution. Some evidence that agnostic learning of several monotone classes is hard is already known and relies on cryptographic assumptions [KKMS08, FGKP09, KS09]. Yet the existing evidence is restricted to the very hard regime when OPT is near 1/2 and does exclude learning with excess error of just 1% that would suffice for most practical applications. We give the first general lower bounds for monotone functions that establish hardness in the low-error regime. We also describe simple and explicit monotone functions that are very close to being resilient.

Finally, we give general tools for analysis of approximate resilience. Such tools might find use for proving new agnostic learning lower bounds.

Acknowledgements

Theorem 1.2 and a special case of Theorem 1.1 for symmetric functions were first derived in V.F.’s collaboration with Pravesh Kothari. We thank Pravesh for his permission to include the result in this work. We thank Justin Thaler for his help in deriving Theorem 1.2 and illuminating discussions on the relationship of our characterization of agnostic learning to the pattern matrix method of Sherstov [She11]. We thank Udi Wieder and Yuval Peres for helpful information about the CycleRun function, and Ryan O’Donnell, Johan Håstad, Rocco Servedio and Jan Vondrák for helpful conversations.

References

- [AH11] Per Austrin and Johan Håstad. Randomly supported independence and resistance. *SIAM Journal on Computing*, 40(1):1–27, 2011. 1.1
- [AM09] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009. 1.1
- [BBL98] A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *Proceedings of FOCS*, pages 408–415, 1998. 1.2, 1.2, 1.2, 3.1, 3.1
- [BDLSS12] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. Minimizing the misclassification error rate using a surrogate convex loss. In *ICML*, 2012. 1.3

- [Bec75] William Beckner. Inequalities in Fourier analysis. *Ann. of Math. (2)*, 102(1):159–182, 1975. [1.2](#), [3.2](#)
- [BFJ⁺94] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262. ACM, 1994. [1.3](#), [2](#)
- [Bon70] Aline Bonami. Étude des coefficients de Fourier des fonctions de $L^p(G)$. *Ann. Inst. Fourier (Grenoble)*, 20(fasc. 2):335–402 (1971), 1970. [1.2](#), [3.2](#)
- [Bro] Daniel G. Brown. How I wasted too long finding a concentration inequality for sums of geometric variables. [C.1](#)
- [BT96] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996. [1.2](#), [1.2](#), [1.2](#)
- [CGH⁺85] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 396–407. IEEE, 1985. [1.1](#)
- [DLSS14] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. The complexity of learning halfspaces using generalized linear methods. In *COLT*, pages 244–286, 2014. [1.3](#)
- [Fel68] W. Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, 1968. [C](#)
- [Fel12] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012. [1.3](#), [2](#)
- [FGKP09] V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. On agnostic learning of parities, monomials and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009. [4](#)
- [FGRW12] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012. [1](#)
- [FK14] V. Feldman and P. Kothari. Agnostic learning of disjunctions on symmetric distributions. *arXiv, CoRR*, abs/1405.6791, 2014. [1.1](#), [1.3](#)
- [FLS11] V. Feldman, H. Lee, and R. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *Journal of Machine Learning Research - COLT Proceedings*, volume 19, pages 273–292, 2011. [1.3](#)
- [FPV13] Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. [2](#)
- [Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. [1](#)
- [IT68] Aleksandr Ioffe and Vladimir Tikhomirov. Duality of convex functions and extremum problems. *Russ. Math. Surv.*, 23, 1968. [1.1](#), [2](#)
- [Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. [1.1](#), [1.3](#)

- [KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *Proceedings of FOCS*, pages 68–80, 1988. 3.4.1, C.2
- [KKMS08] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008. 1, 1.1, 1.3, 1.2, 1.2, 1.3, 4, A, A.1
- [KS07] Adam R Klivans and Alexander A Sherstov. Unconditional lower bounds for learning intersections of halfspaces. *Machine Learning*, 69(2-3):97–114, 2007. 1.3
- [KS09] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci.*, 75(1):2–12, 2009. 4
- [KS10] Adam R. Klivans and Alexander A. Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010. 1, 1.3
- [KSS94] M. Kearns, R. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2/3):115–141, 1994. 1
- [Lov87] László Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50. SIAM, 1987. 2
- [LS11] Philip M. Long and Rocco A. Servedio. Learning large-margin halfspaces with more malicious noise. In *NIPS*, pages 91–99, 2011. 1.3
- [LW95] Michael Luby and Avi Wigderson. *Pairwise independence and derandomization*. Citeseer, 1995. 1.1
- [Man95] Y. Mansour. An $O(n^{\log \log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995. 3.2.1, B
- [MO02] Elchanan Mossel and Ryan O’Donnell. On the noise sensitivity of monotone functions. In *Mathematics and Computer Science II*, pages 481–495. Springer, 2002. 1.2
- [MOS04] E. Mossel, R. O’Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer & System Sciences*, 69(3):421–434, 2004. Previously published as “Learning juntas”. 1.2, 1.2
- [O’D03] R. O’Donnell. *Computational Applications of Noise Sensitivity*. PhD thesis, 2003. 1.2, B
- [O’D13] Ryan O’Donnell. *Analysis of boolean functions*. <http://analysisofbooleanfunctions.org>, 2013. 3.2
- [OS07] R. O’Donnell and R. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007. 1.2
- [OW13] Ryan O’Donnell and Karl Wimmer. Kkl, kruskal-katona, and monotone nets. *SIAM J. Comput.*, 42(6):2375–2399, 2013. 1.2, C
- [OW14] Ryan O’Donnell and David Witmer. Goldreich’s prg: Evidence for near-optimal polynomial stretch. In *Conference on Computational Complexity*, 2014. 1.1, 1.2
- [Sch90] Mark F Schilling. The longest run of heads. *College Math. J*, 21(3):196–207, 1990. C.1
- [Ser01] R. Servedio. On learning monotone DNF under product distributions. In *Proceedings of the Fourteenth Annual Conference on Computational Learning Theory*, pages 558–573, 2001. 1.2

- [She11] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. [1.1](#), [1.3](#), [4](#), [4](#)
- [Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, 30(5):776–780, 1984. [1.2](#)
- [Sim07] H. Simon. A characterization of strong learnability in the statistical query model. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 393–404, 2007. [1.3](#)
- [Szö09] Balázs Szörényi. Characterizing statistical query learning: simplified notions and proofs. In *Algorithmic Learning Theory*, pages 186–200. Springer, 2009. [1.3](#), [2](#)
- [Tal93] M. Talagrand. Isoperimetry, logarithmic Sobolev inequalities on the discrete cube and Margulis’ graph connectivity theorem. *GAF*, 3(3):298–314, 1993. [C.2](#)
- [Tal96] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996. [1.2](#)
- [Val12] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20. IEEE, 2012. [1.2](#), [1.3](#)
- [Wie] Udi Wieder. Tennis for the people ii. <http://windowsontheory.org/2012/11/16/tennis-for-> [1.2](#)
- [Yan05] Ke Yang. New lower bounds for statistical query learning. *Journal of Computer and System Sciences*, 70(4):485–509, 2005. [2](#)

A Extension to Product Distributions

We now outline the extension of our characterization of the SQ complexity of agnostic learning to more general product distributions. Let X be the domain of each individual variable, that is our learning problem is defined over X^n . We will start with symmetric product distributions and let Π be a distribution over X . Let $\mathcal{B} = \{B_0(x), B_1(x), \dots\}$ be the basis obtained via Gram-Schmidt orthonormalization on the basis $1, x, x^2, \dots$ with respect to the inner product $\langle f, g \rangle_\Pi = \mathbf{E}_\Pi[f(x)g(x)]$. By definition we obtain that the polynomial degree of B_i is i (for $i \leq |X| - 1$). As special cases this process gives $\{1, \frac{1-\mu \cdot x}{\sqrt{1-\mu^2}}\}$ basis if $X = \{-1, 1\}$ and $\mu = \mathbf{E}_\Pi[x]$; Legendre polynomials when $X = [-1, 1]$ and Π is uniform; and Hermite polynomials when $X = \mathbb{R}$ and Π is the Gaussian $N(1, 0)$ distribution.

For $S \subseteq [n]$ and a function $t : S \rightarrow \mathbb{N}$ let $\Phi_{S,t}(x) = \prod_{i \in S} x_i^{t(i)}$ and $\Psi_{S,t}(x) = \prod_{i \in S} B_{t(i)}(x_i)$. For a finite X we restrict the range of such t ’s to $[|X| - 1]$. Clearly, Ψ ’s are orthonormal functions relative to the inner product $\langle f, g \rangle_{\Pi^n} = \mathbf{E}_\Pi[f(x)g(x)]$.

We now say that a function g is d -resilient relative to Π^n if for every $S \subseteq [n]$ of size at most d and any function $t : S \rightarrow \mathbb{N}$, $\langle g, \Psi_{S,t} \rangle_{\Pi^n} = 0$. Note that equivalently this can be defined as $\langle g, \Phi_{S,t} \rangle_{\Pi^n} = 0$ for all $S \subseteq [n]$ of size at most d and $t : S \rightarrow \mathbb{N}$.

We say that a Boolean f is α -approximately d -resilient relative to Π^n if there exists a d -resilient $g : X^n \rightarrow [-1, 1]$ such that $\mathbf{E}_{\Pi^n}[|f(x) - g(x)|] \leq \alpha$. In the following discussion functions are over X^n and all norms and inner products relative to Π^n .

We now describe generalizations of Theorems [1.3](#), [1.2](#) and [2.3](#). Let $\mathcal{P}_{d,\ell}$ denote the class of polynomials where each monomial has at most d different variables each of degree at most ℓ ; let $\mathcal{P}_d = \mathcal{P}_{d,\infty}$. Note

that by definition this is the span of $\{\Phi_{S,t}\}_{|S|\leq d, t:S\rightarrow[\ell]}$ but is also equal to the span of $\{\Psi_{S,t}\}_{|S|\leq d, t:S\rightarrow[\ell]}$. For a function f , let $\Delta_{\mathcal{P}_{d,\ell}}(f) = \min_{p\in\mathcal{P}_{d,\ell}} \mathbf{E}_{\Pi^n}[|f(\mathbf{x}) - p(\mathbf{x})|]$ and for a concept class \mathcal{C} , let $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C}) = \max_{f\in\mathcal{C}} \Delta_{\mathcal{P}_{d,\ell}}(f)$.

The polynomial ℓ_1 regression algorithm of Kalai et al. for agnostic learning [KKMS08] applies to this general setting and gives the following bound.

Theorem A.1 ([KKMS08]). *Let \mathcal{C} be a concept class over X^n and fix d and ℓ . There exists a SQ algorithm which for any $\varepsilon > 0$ agnostically learns \mathcal{C} over Π^n with excess error $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})/2 + \varepsilon$ and has complexity $\text{poly}((n\ell)^d, 1/\varepsilon)$.*

Our SQ lower bound can be easily seen to generalize to the following statement.

Theorem A.2. *Let \mathcal{C} be a concept class over X^n closed under renaming of variables and assume that \mathcal{C} contains a k -junta which is α -approximately d -resilient over Π^n . Then any SQ algorithm for agnostically learning \mathcal{C} over Π^n with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$ has complexity of at least $m^{1/3}$, where $m = \mathcal{M}(n, k, d)$. In particular, for any constant $\delta > 0$ and $k = n^{1/2+\delta}$, we have $m = n^{\Omega(d)}$.*

Finally, the duality is also easy to verify in this case.

Theorem A.3. *For $f : X^n \rightarrow \{-1, 1\}$ and $0 \leq d \leq n$ let α denote the ℓ_1 distance of f to the closest d -resilient bounded function. Then $\Delta_{\mathcal{P}_d}(f) = 1 - \alpha$.*

Now the upper bound is $(n\ell)^{O(d)}$ with excess error $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})/2$ and the lower bound is $n^{\Omega(d)}$ with excess error of $\Delta_{\mathcal{P}_d}(\mathcal{C})/2$ (if k is not too large). Therefore tightness depends on how fast $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})$ approaches $\Delta_{\mathcal{P}_d}(\mathcal{C})$ as ℓ grows. Note that if \mathcal{C} contains only functions that depend on at most k -variables then convergence of $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})$ to $\Delta_{\mathcal{P}_d}(\mathcal{C})$ depends only on k (and not on n) and also as long as $\ell = n^{O(1)}$ the bounds are still within a polynomial factor.

Non-symmetric product distributions. Now let the domain be $X_1 \times X_2 \times \dots \times X_n$ and the product distribution be $\Pi = \Pi_1 \times \Pi_2 \times \dots \times \Pi_n$. We first note that the upper bound in Thm. A.1 and the duality hold even if the distribution is not symmetric (that is different variables might have different marginal distributions). Therefore we only need to adapt Thm. A.2 to this setting.

Our lower-bound construction requires closed-ness with respect to renaming of variables. That would not suffice if different variables have different marginal distributions. For example ℓ_1 distance to polynomials clearly depends on the marginal distributions of variables and therefore we can no longer claim that the analogue of $\|f_{S_i} - g_{S_i}\|_1 = \|f - g\|_1$ holds in this setting (as we did in the proof of Lemma 2.1). Therefore we will need an additional assumption. Let S be the set of variables of the optimal (in terms of distance to d -resilience) k -junta. We will assume that for every variable $i \in S$, there are many other variables that have the same marginal distribution as variable i . Specifically, there exists a set $I_i \subseteq [n]$, such that for $j_1, j_2 \in I_i$, $\Pi_{j_1} = \Pi_{j_2}$ and the size of I_i is at least s . In addition, we need \mathcal{C} to be closed under renaming of variables, where a variable that is in I_i is renamed to another variable in I_i .

Now we can construct a family of ordered sets S_1, \dots, S_m (each of size k) such that the intersection of any two sets is at most d , and the i 'th element of each set S_j (recall that we think of S_j as an ordered set) is from I_i . This means that X and Π restricted to variables in S_j (ordered in the same way as they are in S_j) are exactly the same as X and Π restricted to variables in S . This means that the proof of the lower bound in Lemma 2.1 applies to this setting, as before essentially verbatim. The complexity is now determined by the size of the largest family of sets with the property we described. By the same argument as in eq.(1) there exists a family of size:

$$\frac{s^k}{\binom{k}{d} s^{k-d}} = \Omega\left(\left(\frac{sd}{k}\right)^d\right).$$

This family has size $n^{\Omega(d)}$ for $s = n^{\Omega(1)}$ and a large range of parameters k and d (e.g. $d = k^{1-\Omega(1)}$).

B Bound on the low-degree Fourier weight of Tribes

The $\text{Tribes}_{w,s} : \{-1, 1\}^{sw} \rightarrow \{-1, 1\}$ function is the disjunction of s disjoint conjunctions, each of width w . For a set $T \subseteq [n]$ let T_i denote the intersection of T with the variables in the i -th conjunction. We use the following expressions proved in [Man95]:

$$\widehat{\text{Tribes}_{w,s}}(T) = \begin{cases} 2(1 - 2^{-w})^s - 1 & T = \emptyset \\ 2(-1)^{k+|T|} 2^{-kw} (1 - 2^{-w})^{s-k} & k = \#\{i : T_i \neq \emptyset\} > 0 \end{cases} \quad (4)$$

Recall that we write Tribes to denote $\text{Tribes}_{w,s}$ with $s = (\ln 2)2^w$; thus $w \approx \log n - \log n \ln n$ and $s \approx n/(\log n)$.

Proposition B.1. *For any $d \leq w$ the Fourier weight of Tribes on degree d and below is at most*

$$\sum_{|S| \leq d} \widehat{\text{Tribes}}(S)^2 \leq 2 \frac{(2 \ln n)^{2d+4}}{n}.$$

Proof. The proof follows Ryan O'Donnell's thesis, pages 66 – 67 [O'D03]. Using the calculations above, we have that for any $T \subseteq [n]$ with $k = \#\{i : T_i \neq \emptyset\}$:

$$\widehat{\text{Tribes}}(T)^2 \leq \left(\frac{2 \ln n}{n} \right)^{2k}.$$

For any k , the number of coefficients that have degree at most d and intersect k conjunctions is at most

$$\sum_{j=0}^d \binom{s}{k} \binom{kw}{j} \leq (d+1) s^k (kw+1)^d \leq n^k w^{2d+2}.$$

The last inequality holds because $s \leq n$ and $k \leq d$ (and we assume that $d \leq w$). Summing over $1 \leq k \leq d$, we obtain:

$$\begin{aligned} \sum_{|T| \leq d} \widehat{\text{Tribes}}(T)^2 &\leq \sum_{k=1}^d n^k w^{2d+2} \left(\frac{2 \ln n}{n} \right)^{2k} \\ &\leq w^{2d+2} \sum_{k=1}^d \left(\frac{(2 \ln n)^2}{n} \right)^k \\ &\leq 2w^{2d+2} \frac{(2 \ln n)^2}{n} \\ &\leq 2 \frac{(2 \ln n)^{2d+4}}{n}, \end{aligned}$$

where we used $w \leq 2 \ln n$ in the last step. \square

C Proofs concerning CycleRun

To aid us in proving properties of CycleRun, we will require several bounds involving Gaussian approximations. Specifically, we will make use of the functions $f_t : \{-1, 1\}^n \rightarrow \{-1, 0, 1\}$ that appear in [OW13]. We define $|x| = \sum_{i=1}^n x_i$ for a string $x \in \{-1, 1\}^n$. These functions f_t are defined so that

$$f_t(x) = \begin{cases} 1 & \text{if } |x| > t\sqrt{n} \\ 0 & \text{if } -t\sqrt{n} \leq |x| \leq t\sqrt{n} \\ -1 & \text{if } |x| < -t\sqrt{n} \end{cases}$$

We use three properties (implicitly) appearing in [OW13] that follow from error estimates for the Central Limit Theorem [Fel68]: for large enough n and $\sqrt{\log n}/100 < t < n^{1/10}$, we have

$$\phi(t)\sqrt{n}/3 \leq \text{Inf}(f_t) \leq 3\phi(t)\sqrt{n} \quad (5)$$

$$\phi(t)/(3t) \leq \Pr_{\mathbf{x}}[f_t(\mathbf{x}) \neq 0] \leq 3\phi(t)/t \quad (6)$$

$$\Pr[|\mathbf{x}| = t] \leq 4\phi(t)/\sqrt{n}. \quad (7)$$

where ϕ is the probability density function of the standard Gaussian distribution: $\phi(u) = \frac{1}{\sqrt{2\pi}} \exp(-u^2/2)$; and $\text{Inf}(f_t) = \mathbf{E}_{\mathbf{x}}[f_t(\mathbf{x}) \cdot |\mathbf{x}|] = \sum_{i \in [n]} \hat{f}_t(\{i\})$. We note that $\text{Inf}(g) = \mathbf{E}_{\mathbf{x}}[g(\mathbf{x}) \cdot |\mathbf{x}|] = \sum_{i \in [n]} \hat{g}(\{i\})$ for a monotone Boolean function $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$.

Definition C.1. For every $x \in \{-1, 1\}^n$, define the set Shift_x to contain the following:

- $x^\alpha = x_{(1+\alpha \bmod n)} \cdots x_{(n+\alpha \bmod n)}$, for $0 \leq \alpha \leq n-1$.
- $-x^\alpha = -x_{(1+\alpha \bmod n)} \cdots -x_{(n+\alpha \bmod n)}$, for $0 \leq \alpha \leq n-1$.

Note that $|\text{Shift}_x|$ always divides $2n$, and if the Hamming weight of x is relatively prime to n , then $|\text{Shift}_x| = 2n$. Because CycleRun is odd and invariant under cyclic shifts, CycleRun is 1 on exactly half the points of Shift_x .

Theorem 3.7. There exist universal constants c_1, c_2 such that for every $n \geq c_2$, there exists a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ such that:

1. For all $S \subseteq [n]$ such that $|S| \leq 1$, $\hat{f}(S) = 0$, and
2. $\mathbf{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot \text{CycleRun}(\mathbf{x})] \geq 1 - 2c_1 \cdot \sqrt{\frac{\log(n)}{n}}$, which implies $\Pr_{\mathbf{x}}[f(\mathbf{x}) \neq \text{CycleRun}(\mathbf{x})] \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}}$.

Proof. Given $\text{CycleRun} : \{-1, 1\}^n \rightarrow \{-1, 1\}$, we construct a set $\bar{S} \subseteq \{-1, 1\}^n$ using the greedy algorithm $\text{Const}_{\bar{S}}(\text{CycleRun}, n)$ described in Figure 1.

Given the set \bar{S} outputted by $\text{Const}_{\bar{S}}(\text{CycleRun}, n)$, the function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is defined in the following way:

$$f(x) = \begin{cases} \text{CycleRun}(x) & \text{if } x \notin \bar{S} \\ -\text{CycleRun}(x) & \text{if } x \in \bar{S}. \end{cases}$$

Clearly, $\mathbf{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot \text{CycleRun}(\mathbf{x})] \geq 1 - 2c_1 \cdot \sqrt{\frac{\log(n)}{n}}$, since the set \bar{S} satisfies $|\bar{S}| \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$. Additionally, f is clearly balanced due to the structure of the set Shift_x of modified points in each iteration of $\text{Const}_{\bar{S}}$ and the fact that CycleRun is odd. Thus, it remains to show that $\hat{f}(S) = 0$ for all $S \subseteq [n]$ such that $|S| \leq 1$.

$\text{Const}_{\overline{S}}(\text{CycleRun}, n)$

1. Initialize $\overline{S} = \emptyset, \overline{S}' = \emptyset$.
2. Initialize $\sigma = 2^n \cdot \sum_{i \in [n]} \widehat{\text{CycleRun}}(\{i\})$.
3. While $|\overline{S}| \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$, do the following:
 - 3a. Find some x with maximal value of $|x|$ such that $\text{CycleRun}(x) = 1$ and such that $x \notin \overline{S}$.
 - 3b. If $\sigma - 2|\text{Shift}_x| \cdot |x| < 0$, then find an $x^* \notin \overline{S}$ such that $|x^*| = 1$ and $\text{CycleRun}(x^*) = 1$ (if no such x^* exists, exit loop and output “Fail.”). Then set $\overline{S} := \overline{S} \cup \text{Shift}_{x^*}$, set $\overline{S}' = \overline{S}' \cup \text{Shift}_{x^*}$, and set $\sigma := \sigma - 4n$. If $\sigma = 0$, exit the loop.
 - 3c. If $\sigma - 2|\text{Shift}_x| \cdot |x| > 0$, set $\overline{S} := \overline{S} \cup \text{Shift}_x$ and set $\sigma := \sigma - 2|\text{Shift}_x| \cdot |x|$.
4. Return \overline{S} .

Figure 1: Algorithm for constructing a set of points \overline{S} used to define the 1-resilient function f .

Claim C.2. Consider an execution of $\text{Const}_{\overline{S}}$. At the end of the i -th iteration, $1 \leq i \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$, if $\text{Const}_{\overline{S}}$ has not terminated, let \overline{S}^i denote the current set of points in \overline{S} , let σ^i denote the current setting of the variable σ and let f^i denote the following Boolean function:

$$f^i(x) = \begin{cases} \text{CycleRun}(x) & \text{if } x \notin \overline{S}^i \\ -\text{CycleRun}(x) & \text{if } x \in \overline{S}^i. \end{cases}$$

Additionally, we define $\overline{S}^0 = \emptyset$, $\sigma^0 = 2^n \cdot \sum_{i \in [n]} \widehat{\text{CycleRun}}(\{i\})$, and $f^0 = \text{CycleRun}$.

For every $0 \leq i \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$ the following invariants hold:

1. $\widehat{f}^i(\{1\}) = \widehat{f}^i(\{2\}) = \dots = \widehat{f}^i(\{n\})$.
2. $\sigma^i = 2^n \cdot \sum_{j \in [n]} \widehat{f}^i(\{j\})$.
3. $\sigma^i = 4nw \geq 0$, for some integer w .

Proof. Proof by induction.

Base Case: The base case follows trivially from the definition of CycleRun and the definition of \overline{S}^0 , σ^0 , f^0 .

Inductive Case: Assume the invariants hold for all $0 \leq j \leq i < c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$, we show that the invariants must also hold for $i + 1$.

For every $j \in [n]$, let us consider the quantity $2^n (\widehat{f}^i(\{j\}) - \widehat{f}^{i+1}(\{j\}))$. Note that by flipping the value of f^i on the points in the set Shift_x , $\widehat{f}^i(\{j\})$ is reduced by exactly $1/2^n \cdot 4 \cdot \frac{|\text{Shift}_x| \cdot |x|}{2n}$ for each $j \in [n]$ and so we have that $\widehat{f}^{i+1}(\{1\}) = \widehat{f}^{i+1}(\{2\}) = \dots = \widehat{f}^{i+1}(\{n\})$. Moreover,

$2^n \left(\sum_{j \in [n]} \widehat{f}^i(\{j\}) - \sum_{j \in [n]} \widehat{f}^{i+1}(\{j\}) \right) = 2|\text{Shift}_x| \cdot |x|$ and so we have that

$$\begin{aligned} \sigma^{i+1} &= \sigma^i - 2|\text{Shift}_x| \cdot |x| \\ &= 2^n \cdot \sum_{j \in [n]} \widehat{f}^i(\{j\}) - 2|\text{Shift}_x| \cdot |x| \\ &= 2^n \cdot \sum_{j \in [n]} \widehat{f}^{i+1}(\{j\}), \end{aligned}$$

where the second equality holds by the induction hypothesis.

Finally, since $\sigma^{i+1} = 2^n \cdot \sum_{j \in [n]} \widehat{f}^{i+1}(\{j\})$ and f^{i+1} is an odd $\{-1, 1\}$ -valued function, we have that $\sigma^{i+1} = 4nw$ for some integer $w \geq 0$.

□

We proceed to show that $\text{Const}_{\overline{S}}$ terminates. Our goal is to show that at the termination of the algorithm, we have $\sigma = 0$.

Claim C.3. *The algorithm $\text{Const}_{\overline{S}}$ always reaches a point where the condition in line 3b is true.*

Proof. We use the functions f_t from the beginning of this section. Take $t' = \sqrt{\log n - 2 \log \log n - C}$ for a constant C to be determined later. Then $\phi(t') = \frac{1}{2\pi} e^{C/2} (\log n) / \sqrt{n}$, so $\text{Inf}(f_{t'}) \geq \frac{1}{6\pi} e^{C/2} \log n$ and $\Pr_x[f_{t'}(x) \neq 0] \leq \frac{3}{2\pi} e^{C/2} / t' \leq \frac{3}{\pi} e^{C/2} \sqrt{\log n / n}$ by Equations 5 and 6 respectively. We choose C so that $\text{Inf}(f_{t'}) \geq 3 \cdot \text{Inf}(\text{CycleRun})$, which can be done since $\text{Inf}(\text{CycleRun}) = O(\log n)$.

We claim that $\text{Const}_{\overline{S}}$ does not include any strings x in \overline{S} with $3 \leq |x| < t'$ (and thus none with $-t' < |x| \leq -3$). Suppose that this claim is false. Because the algorithm is greedy, then every string x where $\text{CycleRun}(x) = 1$ with $t' \leq |x| \leq n$ is corrupted and in \overline{S} . Since CycleRun is odd and monotone, at least half of the strings where $|x| = k$ are corrupted for $t' \leq k \leq n$. The contribution to be reduction in the first-order Fourier coefficients when we flip the value on these strings from 1 to -1 is at least $(1/2)\text{Inf}(f_{t'}) \geq (3/2)\text{Inf}(\text{CycleRun})$. But this implies that the sum of first-order Fourier coefficients for the corrupted function is at most $-(1/2)\text{Inf}(\text{CycleRun}) < 0$. This implies that $\sigma < 0$ in the execution of $\text{Const}_{\overline{S}}$, which is a contradiction since σ stays nonnegative during the execution of the algorithm.

It remains to show that the condition in line 3 is satisfied throughout the execution of $\text{Const}_{\overline{S}}$. Because no strings with $3 \leq |x| < t'$ or $t' < |x| \leq -3$ are corrupted, the fraction of strings corrupted is at most $\Pr_x[f_{t'}(x) \neq 0] + \Pr_x[|x| = \pm 1] = O(\sqrt{\log n / n})$. Thus at most $c_1 \sqrt{\frac{\log n}{n}} 2^n$ strings are in \overline{S} , so the condition in line 3 holds.

□

Next, we argue that when $\text{Const}_{\overline{S}}$ reaches the point where the condition in line 3b evaluates true, there always exists a point $x^* \notin \overline{S}$ such that $\text{CycleRun}(x^*) = 1$ and $|x^*| = 1$. We first prove two lemmas.

Lemma C.4. *Let S_1^1 be the set of $x \in \{-1, 1\}^n$ such that $|x| = 1$ and $\text{CycleRun}(x) = 1$. Then $|S_1^1| \geq 2n^2$.*

Proof. Note that since CycleRun is odd, we have that $\sum_{x: |x|=\pm 1} \text{CycleRun}(x) = 0$. Moreover, since CycleRun is monotone, we must have that $\sum_{x: |x|=1} \text{CycleRun}(x) \geq \sum_{x: |x|=-1} \text{CycleRun}(x)$. Therefore, we must have that $\sum_{x: |x|=1} \text{CycleRun}(x) \geq 0$. Since CycleRun is $\{-1, 1\}$ -valued, this immediately implies that at least half of the points x where $|x| = 1$ are such that $\text{CycleRun}(x) = 1$. There are $\binom{n}{(n-1)/2} \geq 4n^2$ such strings where $|x| = 1$, so we have that $|S_1^1| \geq 2n^2$. This concludes the proof of Lemma C.4. □

Lemma C.5. $|\overline{S}'| \leq 2n^2$.

Proof. Consider the first time the condition in line 3b evaluates to true. Then there is some x such that $\text{CycleRun}(x) = 1$ and such that $\sigma - 2|\text{Shift}_x| \cdot |x| < 0$. Since $|x| \leq n$, this implies that $\sigma \leq 4n^2$. Moreover, in each iteration $2n$ points are added to \overline{S}' , and σ is reduced by $4n$. Thus, after at most n iterations, σ is reduced to 0. These iterations are the only iterations that contribute to \overline{S}' , so $|\overline{S}'| \leq n \cdot 2n = 2n^2$ as claimed. \square

We proceed to show that when the condition in line 3b is true, there is an $x^* \notin \overline{S}$ such that $\text{CycleRun}(x^*) = 1$ and $|x^*| = 1$. By Lemma C.4, there exist at least $2n^2$ number of points x^* such that $\text{CycleRun}(x^*) = 1$ and $|x^*| = 1$. Thus, if $\text{Const}_{\overline{S}}$ reaches a point where the condition in line 3b evaluates to true and there is no point $x^* \notin \overline{S}$ such that $\text{CycleRun}(x^*) = 1$ and $|x^*| = 1$, then it must be the case that all such x^* are already contained in \overline{S} . But since we have by Lemma C.5 that $|\overline{S}'| \leq 2n^2$ then we must have that some point y such that $\text{CycleRun}(y) = 1$ and $|y| = 1$ was added to \overline{S} before the first time the condition in line 3b evaluates to true. But the first time the condition in line 3b evaluates to true, we must have that $|x| > 1$, and since $\text{Const}_{\overline{S}}$ always chooses to add points y with maximal $|y| \geq |x| > 1$ to the set \overline{S} , this is impossible.

We have now argued that $\text{Const}_{\overline{S}}$ always reaches a point where the condition in line 3b is true, and that whenever this occurs there always exists a point $x^* \notin \overline{S}$ such that $\text{CycleRun}(x^*) = 1$ and $|x^*| = 1$. This immediately implies that when $\text{Const}_{\overline{S}}$ completes, we have $\sigma = 0$ and $|\overline{S}| \leq c_1 \sqrt{\frac{\log n}{n}} 2^n$. As in the beginning of the proof, we take f to be function to be the function such that

$$f(x) = \begin{cases} \text{CycleRun}(x) & \text{if } x \notin \overline{S} \\ -\text{CycleRun}(x) & \text{if } x \in \overline{S}. \end{cases}$$

Clearly, $\Pr_x[f(x) \neq \text{CycleRun}(x)] = |\overline{S}| \leq c_1 \sqrt{\frac{\log n}{n}} 2^n$, and applying the invariants of Claim C.2 shows that f is 1-resilient, concluding the proof of Theorem 3.7. \square

This analysis almost works for any balanced monotone function with influence $O(\log n)$, such as Tribes. While the above could be adapted in a straightforward matter to show that there is a Boolean function close to Tribes with very small constant and first-order Fourier coefficients, showing that all of these Fourier coefficients can be made *exactly* zero seems challenging. Since we are applying these results to juntas, our proofs can not tolerate even exponentially small Fourier coefficients. The structure of CycleRun is quite amenable to “local” changes while retaining structure.

C.1 Influence bound for Cycle Run

The main result of this section is the following:

Theorem C.6. $\text{Inf}(\text{CycleRun}) = O(\log n)$.

The condition on CycleRun given in Definition 3.6 implies that for every influential edge $(x, x^{\oplus i})$, at least one of the endpoints is in the first two cases in Definition 3.6, and the pivotal coordinate i occurs in a maximum length run. Thus $\text{Inf}(\text{CycleRun}) \leq 2 \mathbb{E}_{x \sim \mathcal{U}}[\ell(x) \cdot (r_{\ell(x)}(x) + 1)]$, where $\ell(x)$ is the maximum length run in the string x , $r_i(x)$ is the number of maximal runs of length exactly i in x , and \mathcal{U} is the uniform distribution on $\{-1, 1\}^n$. In this section, we will not consider the runs wrapping around, and the $+1$ here takes care of the case that we “split” the cycle in a maximum length run to lay out the bits in a line.

We make use of a result from [Sch90]:

Theorem C.7. $\mathbb{E}_{x \sim \mathcal{U}}[\ell(x)] = O(\log n)$

Thus $\text{Inf}(\text{CycleRun}) \leq 2 \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}(\mathbf{x})] + O(\log n)$, so the remainder of the section is devoted to showing $\mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}(\mathbf{x})] = O(\log n)$. To aid in our analysis, we will consider different distributions over binary strings. Consider the following method of generating a string $\mathbf{x} \sim \mathcal{U}$:

1. Initialize \mathbf{x} to the empty string, and set b to a uniform ± 1 random bit \mathbf{b} .
2. (Iterative step) Assuming there are still $j > 0$ bits of \mathbf{x} to determine, then draw $\mathbf{g} \sim \text{Geometric}(1/2)$ conditioned on \mathbf{g} being at most j , and set the next \mathbf{g} bits of \mathbf{x} to b .
3. If not all n bits of \mathbf{x} are set, set b to $-b$ and return to step 2.
4. If all bits of \mathbf{x} are set, then \mathbf{x} is a uniformly random string in $\{-1, 1\}^n$.

Further, if we want to condition on the maximum run in \mathbf{x} being at most some value t , we can replace the conditioning in step 2 from “being at most j ” to “being at most $\min\{t, j\}$ ”.

Lemma C.8. *For $\mathbf{g} \sim \text{Geometric}(1/2)$, and $1 \leq g \leq t$, we have $\Pr[\mathbf{g} = g | \mathbf{g} \leq t] \leq 2\Pr[\mathbf{g} = g]$.*

Proof. Follows directly from conditional probability and the fact that $\Pr[\mathbf{g} \leq t] \geq 1/2$ for all $t \geq 1$. \square

For an integer $k > 0$, we define the distribution \mathcal{G}_k on binary strings of varying length such that a draw from \mathcal{G}_k is $\mathbf{b}^{g_1}(-\mathbf{b})^{g_2}\mathbf{b}^{g_3} \dots \mathbf{b}^{g_k}$ if k is odd and $\mathbf{b}^{g_1}(-\mathbf{b})^{g_2}\mathbf{b}^{g_3} \dots (-\mathbf{b})^{g_k}$ if k is even. Here, the \mathbf{g}_i ’s are independent $\text{Geometric}(1/2)$ variables, and \mathbf{b} is a uniform ± 1 bit.

Lemma C.9.

$$\mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}(\mathbf{x}) | \ell(\mathbf{x}) = t] \leq t(2^{1-t}n + 1)$$

Proof. We first claim that

$$\mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}(\mathbf{x}) | \ell(\mathbf{x}) = t] \leq t + \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_t(\mathbf{x}) | \ell(\mathbf{x}) \leq t]$$

To see this, note that if we further condition on the first run of length t selected, this expectation is maximized when the first run is of length t . Also, the expectation can only increase if we allow all n more bits to be set rather than $n - t$. Since the first run is of length t , we only need the maximum length run to be at most t in the rest of the string.

Now we have

$$t + \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_t(\mathbf{x}) | \ell(\mathbf{x}) \leq t] \leq t + t \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[r_t(\mathbf{x}) | \ell(\mathbf{x}) \leq t] \leq t + t \mathbf{E}_{\mathbf{y} \sim \mathcal{G}_n}[r_t(\mathbf{y}) | \ell(\mathbf{y}) \leq t]$$

where the second inequality comes from the fact that \mathbf{x} is generated by at most n runs, and not bounding the length of the string only increases the possible number of runs of length t , conditioned on the maximum length run being at most t . By Lemma C.8, the probability of a single run being of length t is at most 2^{1-t} , so we have

$$t + t \mathbf{E}_{\mathbf{y} \sim \mathcal{G}_n}[r_t(\mathbf{y}) | \ell(\mathbf{y}) \leq t] \leq t + t(2^{1-t}n) = t(2^{1-t}n + 1)$$

completing the proof. \square

Lemma C.10.

$$\Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \leq t] \leq (1 - 2^{-t})^{n/8} + \exp(-n/32)$$

Proof. For $\mathbf{x} \in \{-1, 1\}^n$, let $\text{runs}(\mathbf{x})$ be the number of runs in \mathbf{x} . We first show that with probability at least $1 - \exp(-n/32)$, a string $\mathbf{x} \sim \mathcal{U}$ has $\text{runs}(\mathbf{x}) \geq n/8$. To do this, we prove that with probability $1 - \exp(-n/32)$, the first $n/8$ runs of \mathbf{x} contain at most $n/2$ bits. Note that we may instead bound the number of bits in $\mathbf{y} \sim \mathcal{G}_{n/8}$, since each run of $\mathcal{G}_{n/8}$ can only be longer.

The expected number of bits in $\mathcal{G}_{n/8}$ generated is $n/4$, and this number of bits is concentrated around its mean; the number of bits has a negative binomial distribution. By [Bro], we have

$$\Pr_{\mathbf{y} \sim \mathcal{G}_{n/8}}[\text{bits}(\mathbf{y}) > 2(n/4)] \leq \exp(-n/32)$$

where the second inequality holds because the number of runs does not increase the probability of getting a longer run, and the distributions of the lengths of each run in \mathbf{x} are identical to (or conditioned on being shorter than) the lengths of the runs in $\mathcal{G}_{n/8}$. We then have:

$$\begin{aligned} \Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \leq t] &\leq \Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \leq t, \text{runs}(\mathbf{x}) \geq n/8] + \exp(-n/32) \\ &\leq \Pr_{\mathbf{y} \sim \mathcal{G}_{n/8}}[\ell(\mathbf{y}) \leq t] + \exp(-n/32) \end{aligned}$$

where the second inequality holds because the length of each run of \mathbf{x} is distributed identically (or conditioned to be shorter) to each run of \mathbf{y} , and considering fewer runs only decreases the chances of obtaining a run longer than t . It is then straightforward to calculate $\Pr_{\mathbf{y} \sim \mathcal{G}_{n/8}}[\ell(\mathbf{y}) \leq t] = (1 - 2^{-t})^{n/8}$, since $\Pr[\mathbf{g} \leq t] = 1 - 2^{-t}$ for $\mathbf{g} \sim \text{Geometric}(1/2)$. \square

We now proceed to show $\mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}] = O(\log n)$, starting by applying total expectation and applying Lemma C.9:

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell(\mathbf{x})}(\mathbf{x})] &= \sum_{t=1}^n \Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) = t] \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) \cdot r_{\ell}(\mathbf{x}) = t] \\ &\leq \sum_{t=1}^n \Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) = t] t (2^{1-t} n + 1) \\ &\leq \mathbf{E}_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x})] + \sum_{t=1}^n \Pr_{\mathbf{x} \sim \mathcal{U}}[\ell(\mathbf{x}) = t] t 2^{1-t} n \\ &\leq O(\log n) + \sum_{t=1}^n ((1 - 2^{-t})^{n/8} + \exp(-n/32)) t 2^{1-t} n \\ &\leq O(\log n) + \sum_{t=1}^n (1 - 2^{-t})^{n/8} t 2^{1-t} n \\ &\leq O(\log n) + \sum_{t=1}^n t n 2^{1-t} \exp(-2^{-t} n/8) \end{aligned}$$

Letting $a_t = t n 2^{1-t} \exp(-2^{-t} n/8)$, we see that $a_{t-1}/a_t < 3/4$ when $2 \leq t \leq \log n - 10$, and $a_{t+1}/a_t < 3/4$ when $\log n + 10 \leq t \leq n$. Also, $a_t \leq O(\log n)$ for each term where $\log n - 10 \leq t \leq \log n + 10$. So the proof is completed by noting the above is at most

$$\begin{aligned}
& O(\log n) + \sum_{t=2}^{\log n-10} a_{\log n-10} (3/4)^{\log n-10-t} + \sum_{t=\log n-9}^{t=\log n+9} a_t + \sum_{t=\log n+10}^n a_{\log n+10} (3/4)^{t-(\log n+10)} \\
& \leq O(\log n) \left(\sum_{t=2}^{\log n-10} (3/4)^{\log n-10-t} + \sum_{t=\log n-9}^{t=\log n+9} 1 + \sum_{t=\log n+10}^n (3/4)^{t-(\log n+10)} \right) = O(\log n).
\end{aligned}$$

C.2 Lower bound for monotonicity-resiliency distance

We give a lower bound for distance between monotonicity and resiliency that matches the bound for CycleRun up to constant factors.

Theorem C.11. *For every monotone function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ and 1-resilient $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$, we have $\Pr_{\mathbf{x}}[f(\mathbf{x}) \neq g(\mathbf{x})] \geq \Omega(\sqrt{\frac{\log n}{n}})$.*

Proof. If $\text{Var}[f] < 1/2$, then $\widehat{f}(\emptyset)^2 > 1/2$, and $\Pr[f \neq g] \geq \frac{1}{4}E[(f-g)^2] \geq 1/8$ for any balanced (hence 1-resilient) Boolean function g . If $\widehat{f}(\{i\}) > n^{-0.49}$ for some i , then f is $\Omega(n^{-0.49})$ -far from every Boolean function g where $\widehat{g}(\{i\}) = 0$.

We assume $\text{Var}[f] \geq 1/2$ and $\widehat{f}(\{i\}) \leq n^{-0.49}$ for all $i \in [n]$. Since f is monotone, $\text{Inf}_i(f) \leq n^{-0.49}$ for all $i \in [n]$, and by (Talagrand's strengthening of) the KKL Theorem [Tal93, KKL88], $\text{Inf}(f) \geq K \log n$ for some constant K , and $\sum_{i \in [n]} \widehat{f}(\{i\}) \geq K \log n$. Let $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a 1-resilient Boolean function; we will show that $\Pr_{\mathbf{x}}[f(\mathbf{x}) \neq g(\mathbf{x})] = \Omega(\sqrt{\frac{\log n}{n}})$.

Recall the functions f_t defined earlier:

$$f_t(x) = \begin{cases} 1 & \text{if } |x| > t\sqrt{n} \\ 0 & \text{if } -t\sqrt{n} \leq |x| \leq t\sqrt{n} \\ -1 & \text{if } |x| < -t\sqrt{n} \end{cases}$$

Select t to be the largest t such that f_t satisfies $\Pr[f_t(\mathbf{x}) \neq 0] \geq \Pr[(f-g)(\mathbf{x}) \neq 0] = \Pr[f(\mathbf{x}) \neq g(\mathbf{x})]$. We then have $K \log n \leq \sum_{i \in [n]} \widehat{f_t}(\{i\}) \leq \sum_{i \in [n]} \widehat{f}(\{i\})$, where the second inequality holds because f_t maximizes the sum of the linear coefficients for any function with support size $\Pr[f_t(\mathbf{x}) \neq 0]$, and the support size of f_t is at least the support size of $f - g$.

Again, because f_t is monotone, $\text{Inf}(f_t) = \sum_{i \in [n]} \widehat{f_t}(\{i\})$. Equation 5 implies that $(3K \log n)/\sqrt{n} \geq \phi(t) \geq (K \log n)/(3\sqrt{n})$, and it follows that $t \leq 4\sqrt{\log n}$. From Equation 6, we have $\Pr_{\mathbf{x}}[f_t(\mathbf{x}) \neq 0] \geq (4K/3)\sqrt{\frac{\log n}{n}}$. By the choice of t , we have

$$\begin{aligned}
\Pr_{\mathbf{x}}[f(\mathbf{x}) \neq g(\mathbf{x})] & > \Pr_{\mathbf{x}}[f_{t+1}(\mathbf{x}) \neq 0] \\
& \geq \Pr_{\mathbf{x}}[f_t(\mathbf{x}) \neq 0] - 2\Pr_{\mathbf{x}}[|\mathbf{x}| = t] \\
& \geq \frac{4K}{3}\sqrt{\frac{\log n}{n}} - 24K\frac{\log n}{n} = \Omega\left(\sqrt{\frac{\log n}{n}}\right),
\end{aligned}$$

where the first inequality is an application of the union bound, and the second is an application of Equation 7. \square